Statistical Mechanics Approach to Inverse Problems on Networks

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Fisica - XXVIII ciclo

Tesi di dottorato

Statistical Mechanics Approach to Inverse Problems on Networks

Alessandro Ingrosso

Relatore:
Prof. Riccardo Zecchina

Coordinatore:
Prof. Arianna Montorsi

A mamma e papá $\forall$ *cosa.*

Contents

Acknowledgements

I'd like to express my acknowledgments to Riccardo Zecchina for the being constantly supportive and encouraging, for interesting discussions and help, and for giving me great opportunities.

I wish to thank Carlo Baldassi and Alfredo Braunstein for their daily help, for being never tired of teaching me something new, even unconsciously. Marco Zamparo was a window open onto the vast realm of mathematical rigour. More importantly, he is the nicest guy. I enjoyed working together with Luca Saglietti and Carlo Lucibello and I hope I'll continue to do that in the future. Thanks to Luca dall'Asta, Andrea Pagnani, Carla Bosia and all the people in the CMP lab.

Apart from these pretty standard PhD thesis acknowledgments, thank you mom, thank you dad, ultimately the only important people when it comes to the crunch. The rest is thermal fluctuations.

CHAPTER 1

Introduction

While the forward problem in Physics consists in predicting the outcome of an experiment given a complete description of a system, inverse problems consist in using the result of some measurements to infer the values of the parameters that characterize a system, or to build a model of a system from a (generally limited) number of observations. At least in the context of deterministic models in Physics, the forward problem has a unique solution. In inverse problems, there is in general no unique solution, so that a probabilistic formulation has to be contemplated, where some extent of a priori information on the parameters, together with the structure of the model itself, has to be incorporated in the Bayesian prior. One then comes up with the hard problem of computing average quantities over a complicated posterior probability distribution, which generally involves a large number of variables.

In this Thesis, I will concentrate on inverse problems that are defined on different kinds of networks, and show how methods borrowed from the Statistical Physics of Disordered Systems, Spin Glasses in particular, can be effectively used not just for the purpose of theoretical analysis, but also to construct practical inference techniques based on suitable approximation schemes. The network structure is strongly reflected in that of the corresponding probabilistic graphical model one builds for inference purposes: simple variables interact with mutual local dependencies, emerging from the factorization properties of the joint probability distribution. The effectiveness of Statistical Physics methods in inference stems from the fact that most problems, when studied at equilibrium, can be formulated in the framework of factor graphs, namely bipartite graphs composed of variable nodes connected via factor nodes, that represent (physical) interactions. Hence, the analysis of large scale probabilistic models poses similar problems as one encounters in Physics trying to analyze the equilibrium properties of systems involving a large number of interacting particles.

The theory of Spin Glasses is a corpus of mathematical methods and concepts that tries to describe the behavior of certain kinds of magnetic materials whose inner structure is characterized by disorder, because of impurities in random locations. The relevance of the methods developed in the study of Spin Glasses, such as Replica and Cavity methods [**1**] goes beyond the realm of purely physical applications. One important example is the connection between the Bethe approximation and the development of Message Passing algorithms that, together with their heuristic extensions, can be used to analyze and solve hard problems which appears in error correcting codes, Bayesian inference in complicated graphical models, Constraint Satisfaction Problems and much more, as I will extensively discuss in this work.

Message Passing algorithms operate on messages that are associated with edges of the factor graph, through recursive updates carried out only locally at the vertices of the graph [**2**]. I will focus on Belief Propagation, a method that can be proven to compute the exact marginals of a probability distribution associated to a tree-structured factor graph. It is very interesting to note that Belief Propagation was found to perform very well on general loopy graphs as well, provided that the underlying graph is locally a tree.

In this Thesis, I will deal with two main topics: the first section, composed of chapters 2, 4 and 5, is devoted to the problem of learning in neural networks with discrete synaptic weights. Deep learning is, without any doubt, the most interesting field in contemporary machine learning, and has experienced a boost since the first effective learning methods have been developed in Deep Belief Nets [**3**]: deep neural networks are now considered method of choice for a variety of hard classification tasks, especially in the field of image recognition [**4**]. Nevertheless, a general theory of the generalization properties of deep networks is still lacking, the nature of deep representations is debated and little is known about the precise role of unsupervised pre-training, fine tuning, as well as optimality of Back Propagation in a deep setting [**3, 5, 6**].

The study of the generalization abilities of neural networks has a long tradition in the history of Statistical Mechanics: since physicists realized the usefulness of the Spin Glass formalism in neural network models, methods in equilibrium and non-equilibrium Statistical Physics have been used to deal with various problems, from the investigation of the memory capacity of the perceptron and simple feed-forward neural networks [**7, 8, 9**], to the study of the dynamics of learning algorithms [**10, 11**]. On the other hand, the Physics of disordered magnetic materials has been the source of inspiration for a variety of attractor neural network models of associative memory [**12**], all stemming from the seminal work of J. Hopfield [**13**].

Here, I will focus on a slightly different perspective: the general learning problem in neural networks can be framed as an inverse problem even in the supervised case, i.e. where a prescribed output (in most cases a semantic category) is associated to each pattern. The problem of learning an input-output association is clearly that of inferring a rule. There is a substantial literature about the generalization ability of neural classifiers, the idea being that of identifying the rule that has to be discovered as a given "teacher" network, which produces a set of input-output associations: the task for a "student" network is to solve the inverse problem of discovering the rule from a given fixed set of example pairs. When phrased as a Statistical Mechanics problem (in a way that will be more clear in chapter 4), the interesting phenomenon of discontinuous learning emerges: in the thermodynamic limit, the teacher eventually dominates the measure over all the possible student networks, so that it will be possible to discover it perfectly with probability 1, provided the number of input-output examples exceeds a given threshold, which is specific for each network structure [**14**].

The research program that motivated the first section of this Thesis is that of relating the known analytical results about the learning and generalization properties of simple neural structures to those of the actual solutions that can be found with the available learning methods. This program eventually led to the study of the fine structure of the space of solutions of the Perceptron Learning Problem with binary synapses, a know NP-hard problem for which no provably convergent algorithm is known, but a handful of BP-inspired learning protocols exist, whose properties can be understood when learning is formulated as a Physics problem. The practical relevance of Belief Propagation (and BP-inspired methods) is deeply connected to the structure of the space of solutions: an apparent discrepancy arises between algorithmic performance and the standard analysis of typical solutions. Standard equilibrium Replica calculations describe a structure characterized by strongly isolated typical solutions, which would be hard if not impossible to find by known BP-inspired algorithms. This issue was resolved with the introduction of a Replica based large deviation analysis, finding that dense subdominant clusters, namely regions in the solution space characterized by a high local density of solutions, are the main targets of effective heuristic algorithms. The out of equilibrium analysis of subdominant states correctly identifies the algorithmic critical capacity, the disappearance of such states in high dimension being the main responsible for the transition that is observed in algorithmic performance. Moreover, it interpolates between the typical single solution and the full Bayesian one.

In an effort to construct an algorithmic counterpart of the theoretical analysis in the case of the perceptron, a new Markov Chain Monte Carlo method was introduced which explicitly looks for regions with a high density, or local entropy, of solutions. This entropy based scheme, which I will elaborate on in chapter 5, can be shown to work well even in the completely different setting of random K-satisfiability, a prototypical NP-complete Random Constraint Satisfaction Problem which has been the subject of intense study in the Spin Glass community in recent years.

In the second section, composed of chapters 6, 7 and 8, I will switch to a completely different problem, namely that of irreversible spreading processes on networks. In particular, I will first focus on the problem of inferring the source of an epidemic spreading in a network from very limited observations, namely a single snapshot of the state of the network at a given time. There are a variety of contexts where the underlying graph is known a priori with a fair extent of completeness, as for networks of computers. Consider, for instance, the case of an artificial virus that spreads in such a network, starting from a very limited number of initial nodes: even if it is in principle easy to track the connections between nodes in the network, a user will typically be aware of the infection only at a later time, and the actual spreading path will be unknown. The situation is similar in the case of the spreading of a rumor on a social network. In order to discover the source of the infection, one has to devise a suitable parametrization for the spreading dynamics, and then be able to trace over all the possible paths which are compatible with some given observations.

Here, I consider the case of a single observation at a given time. My work has been focused on the generalization of a previously developed Bayesian approach [**15**] to the scenario of uncertain observations. As I will point out in the following, the inference machinery is quite general, and the particular appeal of the method stems from the fact that the corresponding factor graph (which incorporates the structure of the generative model for the dynamics and it's at the basis of the Message-Passing procedure) is an enriched dual version of the original graph: this implies that the proposed method provides the exact Bayesian solution in the case of tree networks, and it will be shown to be very effective in general graphs with loops, a notable property of the Belief Propagation approximation that proved very useful in a variety of settings, both in inference and combinatorial optimization problems.

I will show how the inference technique can be extended so as to take into account a general continuous time model for the spreading dynamics. More interestingly, a slight modification of the method will be shown to be able to solve a way more general inverse problem, where the network itself is unknown and its structure - together with the weights of each edge - has to be inferred from a limited number of observations coming from different cascades of activation.

Outline of the Thesis

This thesis is organized as follows. In the first two chapters, I will introduce the main theoretical tools, rooted in Statistical Mechanics, that will be used throughout the work, namely Spin Glass theory and Belief Propagation.

The second chapter consists of a very concise introduction to the Replica method and its relation to the theory of neural computation. The Replica method was originally introduced in the computation of the self averaging free energy in the Sherrington-Kirkpatrick model [**16**]. The Spin glass formalism has proven extremely fruitful since the seminal work by E. Gardner [**17**], who introduced the concept of critical capacity in the study of the simplest neural classifier, the perceptron.

The third chapter reviews Belief Propagation, a complementary algorithmic approach which will be used throughout the entire Thesis: the Bethe approximation will be discussed for trees and general factor graphs together with some known results on its convergence properties.

The fourth chapter is devoted to the analysis of the learning problem in perceptrons. The fine structure of the space of solutions of the classification and generalization problem will be investigated, in order to relate the typical case scenario described by standard Replica calculation to the actual properties of BP-based learning algorithms. A new large deviation formalism will be introduced and the analytical results will be supported by extensive simulations. Moreover, I will show that, when used in conjunction with unsupervised pre-training, BP-inspired on-line algorithms can be adopted to train feed-forward networks with binary synaptic weights that perform very well in the case of random input-output associations as well as on benchmarks datasets.

In chapter 5 I will show how the analytical method introduced in the case of the perceptron can be turned into a solver with the introduction of a Monte Carlo approach based solely on the local entropy, namely an estimate of the number of zero energy configurations at a given distance from a reference one.

Chapter 6 is devoted to the patient zero problem in epidemic spreading over networks, a hard inference problem which consists in discovering the source of the spread of an epidemic from single snaphots of the state of a network. I will describe a previously proposed inference method built on a Belief Propagation approximation, that will be later generalized to take into account possible noise in observations as well as ignorance of epidemic parameters. The method will be further generalized in chapter 7 so as to deal with the more realistic continuous-time case.

In the eighth chapter I will give a brief account of a very recent extension of the method discussed in chapter 6 to the problem of network reconstruction from dynamical data. It will be shown how, in the presence of limited observations from a number of cascades of activation over an unknown network, the formalism introduced for solving the patient zero problem can be used to reconstruct the topological structure of the graph. Its effectiveness will be documented by means of simulations on random graphs as well as real social networks.

Conclusions will be drawn in the ninth and last chapter.

CHAPTER 2

From Spin Glasses to Neural Networks

The Replica formalism, initially introduced in the context of Statistical Mechanics for the study of Spin Glasses, proved very helpful in addressing the problem of learning capabilities of neural networks. The first application of this Statistical Mechanics approach appeared in E. Gardner's famous paper [**17**], in which the Replica technique was used to compute the critical capacity of the perceptron with continuous synaptic weights. From there on, various variants of one and multi-layer perceptrons have been studied in the same way: I will briefly sketch the methods and results in a later section.

Very interestingly, it is also possible to construct a generalized Statistical Mechanics formulation of the learning problem of a neural network [**14**], associating to each learning rule a particular Energy function $E(\boldsymbol{J})$, taking value in the weight space of the network: in all cases where $E(\boldsymbol{J})$ has a unique minimum, it is possible to follow a (relatively) simple algorithmic method for the analytical study of the generalization properties of a given learning rule.

In the following sections, I will firstly give a brief introduction to the Replica Method, I will then describe the main approach for the study of the perceptron and more complex network architectures.

2.1. Introduction to Spin Glasses and Replica Theory

A Spin Glass is a model of a disordered magnetic material, in which interactions among spins are randomly distributed: frustration of interactions results in a variety of interesting features, from the exponential number of metastable states to ergodicity breaking and ultrametricity [**1**]. The first model of a Spin Glass was the Edwards-Anderson model [**18**], defined by a d-dimensional spin lattice and nearest neighbors interactions with the following Hamiltonian:

$$H = -\sum_{\langle ij \rangle} J_{ij} S_i S_j \tag{2.1.1}$$

This model can be analytically solved showing, for sufficiently low temperature, a so-called Glass Phase, characterized by vanishing magnetization but non vanishing Edward-Anderson order parameter $q^2 = \frac{1}{N}\sum_i \langle S_i \rangle \langle S_i \rangle$. The most important model in the development of the Replica method was the Sherrington-Kirkpatrick model [**16**], introduced in 1975 and finally solved by Parisi in 1979 [**19**]. It is a mean field model of a Spin Glass, with infinite range interaction prescribed by the Hamiltonian

$$H = -\sum_{i<j} J_{ij} S_i S_j \tag{2.1.2}$$

Clearly, the model is not completely defined until one has given the distribution of the couplings J_{ij}: it is common practice to take them independently distributed according to a gaussian distribution (with a proper scaling for the variance).

In the theory of Spin Glasses, one can distinguish between two different types of calculations of the free energy:

- *Annealed* disorder

$$-\beta N f = \log\langle\langle Z\rangle\rangle \tag{2.1.3}$$

- *Quenched* disorder

$$-\beta N f = \langle\langle \log Z\rangle\rangle \tag{2.1.4}$$

where $\langle\langle\rangle\rangle$ stands for the average over the disorder. The *Annealed* calculation provides an upper bound for the more correct *Quenched* calculation, in which the interaction disorder is kept fixed and the thermal disorder, acting on a faster time scale, is averaged.

The Replica Method stems from an *ad hoc* trick for turning the average of $\log Z$ in the free energy to a power of the partition function Z, by means of the simple relation:

$$\log Z = \lim_{n\to 0} \frac{Z^n - 1}{n} \tag{2.1.5}$$

One can thus rewrite Eq. (2.1.4) as:

$$-\beta N f = \lim_{n\to 0} \frac{\langle\langle Z^n\rangle\rangle - 1}{n} \tag{2.1.6}$$

and consider the nth power of the partition function as the product of partitions functions of n non-interacting replicas of the same system: one has to be very careful in this approach, noting that in principle it is not possible to perform the limit $n \to 0$ with an integer number of Replicas! Let us set the calculation for the Sherrington-Kirckpatrick free energy:

$$\langle\langle Z^n\rangle\rangle = \langle\langle \sum_{\{S_i^a\}} \exp\left(\beta \sum_{a,i<j} J_{ij} S_i^a S_j^a\right)\rangle\rangle \tag{2.1.7}$$

Without going into the details of the calculations, let us point out that the main advantage of such form is the possibility to take first the average over the disorder, and then use an inverse gaussian transformation (*Hubbard-Stratonovich transformation*) to uncouple the spin S_i^a and perform the sum over the configurations: that naturally introduces a coupling of the replicas, that can be handled with the introduction of *overlap* order parameters, defined as:

$$q^{ab} = \frac{1}{N}\sum_i S_i^a S_i^b, \quad a < b \tag{2.1.8}$$

These parameters measure the typical overlap among the configurations in a given state belonging to different replicas, and will be very important in the following discussions. It is then possible to write Eq. (2.1.7) as an integral over the overlap parameters as well as their conjugated fields:

$$\langle\langle Z^n\rangle\rangle = \int \prod_{a<b} dq^{ab} d\hat{q}^{ab} e^{-NG(q^{ab},\hat{q}^{ab})} \tag{2.1.9}$$

where the function in the exponent is defined as

$$G(q^{ab},\hat{q}^{ab}) = -n\frac{\beta^2}{4} - \frac{\beta^2}{2}\sum_{a<b} q^{ab} + i\sum_{a<b} q^{ab}\hat{q}^{ab} - \log\left[\sum_{\{S^a\}} e^{i\sum_{a<b}\hat{q}^{ab}S^aS^b}\right] \tag{2.1.10}$$

This integral can be calculated by the Saddle Point Method: the main difficulty is then to solve the saddle point equations with respect to all the relevant integration variables, finding an extremum for Eq. (2.1.10). The Replica Symmetric Ansatz consists in choosing $q^{ab} = q$, $\hat{q}^{ab} = \hat{q}$, thus restricting oneself to a subspace of the order parameters. Unfortunately, one discovers that the free energy obtained by the RS Ansatz is not a decreasing function of the temperature: this is because of the instability of the RS saddle point, as implied by the eigenvalues of the Hessian Matrix of $G(q^{ab},\hat{q}^{ab})$.

The first Replica Symmetry Breaking calculation was performed by G. Parisi in 1979 [**19**]: in the general r-step RSB, it consists in dividing the n replicas into n/m groups of m replicas each, and introducing a series of overlap parameters $q_0, q_1, ...q_r$ (the same is done for the $\hat{q}^{ab}$ parameters). Consider, for example, a 1-RSB calculation: one would have $q^{ab} = q_1$ if a and b belong to the same group and $q^{ab} = q_0$ if a and b belong to different groups. The structure of the overlap matrix is exemplified in figure 2.1.1.

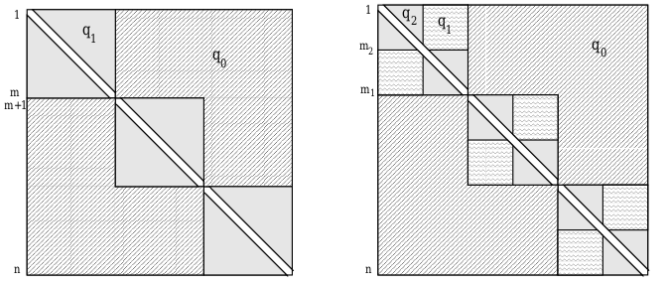

FIGURE 2.1.1. Structure of the q^{ab} matrix with Replica Symmetry Breaking. *Left*: 1RSB. *Right*: 2RSB.

The overlap matrix encodes many interesting geometrical properties of the states of the systems: overlaps between replicas can be directly linked to overlaps between states. Let us introduce the distribution of overlaps for a given disordered system:

$$P_J(q) = \frac{1}{Z^2}\sum_{\boldsymbol{s},\boldsymbol{\tau}} \exp\left[-\beta H_J(\boldsymbol{s}) - \beta H_J(\boldsymbol{\tau})\right]\delta\left(\sum_i s_i\tau_i - q\right) \tag{2.1.11}$$

$$P(q) = \langle\langle P_J(q)\rangle\rangle \tag{2.1.12}$$

By means of a variant of the standard Replica Trick one can express the moments of the distribution (2.1.12) in terms of powers of quantities q^{ab}, leading to the general expression:

$$P(q) = \lim_{n\to 0}\frac{1}{[n(n-1)/2]}\sum_{\{a,b\}}\delta(q^{ab} - q) \tag{2.1.13}$$

$P(q)$ is nothing but the fraction of the elements q^{ab} taking the value q. This is extremely important in view of a geometrical characterization of Replica Symmetry Breaking: the Ansatz is linked to the unconnectedness and clustering of the states of the systems.

In the 1-RSB scheme, for example, one has $P(q) = m\delta(q - q_0) + (1 - m)\delta(q - q_1)$, where q_1 is the typical intra-cluster overlap, and q_0 the typical inter-cluster overlap.

In the following discussion, the phenomenon of Replica Symmetry Breaking will play a major role in the study of the solution space (also known as *version space*) for the learning problem in the perceptron.

2.2. The Gardner Approach to the Storage Problem

Perceptron is the simplest kind of neural model and it serves, in its variants, as a building block for multi-layered networks with complex topologies. Basically, it is a linear discriminator between inputs: consider a set of pattern vectors $\xi^\mu \in \mathbb{R}^N$ and suppose you want to construct a function that discriminates among them, associating to each one a binary output σ^μ, say for simplicity $\sigma^\mu \in \{-1, +1\}$. Here, I will consider a perceptron defined by the vector of synaptic weights $W \in \mathbb{R}^N$ and a sign activation function, with an output o^μ:

$$o^\mu = \text{sign}\,(W \cdot \xi^\mu) \tag{2.2.1}$$

Many learning rules have been developed to deal with the perceptron learning problem, from the classical Hebb and Rosenblatt Rule to Pseudo-Inverse and Adaline Rule [**14**]. Here, I will introduce a method of analysis that is independent from a specific learning rule, and provides information on the structure of the subspace of solutions to a given learning problem.

The Gardner approach is based on the calculation of the typical volume of the *version space*, the subset of weight vectors W compatible with a given learning task, defined by random association between input vectors and outputs, as prescribed by a given probability distribution. Consider then $p = \alpha N$ input vectors independently chosen with probability:

$$p\,(\xi^\mu) = \prod_{i=1}^{N} \left[\frac{1}{2}\delta\,(\xi_i^\mu + 1) + \frac{1}{2}\delta\,(\xi_i^\mu - 1) \right] \tag{2.2.2}$$

Analogously the outputs σ^μ will be chosen indipendently in the set $\{-1, +1\}$ with equal probability. Obviously $\|\xi^\mu\|^2 = N$, and I will add a constraint on the length of the weight vectors, specifically $\|W\|^2 = N$, so that they are randomly chosen in the N sphere.

In order to get the desired associations $o^\mu = \sigma^\mu$, the following conditions have to be fulfilled:

$$\frac{1}{\sqrt{N}}\sigma^\mu W \cdot \xi^\mu \geqslant \kappa \qquad \forall \mu = 1, ..., p \tag{2.2.3}$$

Note that the *stability parameter* $\kappa > 0$ is particularly important in the context of attractor neural network [**12**], in which it is associated to the size of basins of attractions: the conditions of Eq. (2.2.3) may indeed be used to study the retrieval properties of a fully connected Hopfield Model [**20, 21**].

Now, we can calculate the volume in the weight space by means of an integration of an *indicator function*, in the following way:

$$\Omega\left(\xi^{\mu},\sigma^{\mu}\right)=\int d\mu(W)\prod_{\mu=1}^{p}\theta\left(\frac{\sigma^{\mu}}{\sqrt{N}}W\cdot\xi^{\mu}-\kappa\right), \tag{2.2.4}$$

$$d\mu\left(W\right)=\prod_{i=1}^{N}dW_{i}\delta\left(\|W\|^{2}-N\right) \tag{2.2.5}$$

where I have introduced the measure of integration $d\mu\left(W\right)$ according to the spherical constraint. Eq. (2.2.4) is clearly a random quantity, depending on the random variables ξ^{μ} and σ^{μ}. In order to obtain its *typical* value, one can rely on a standard Statistical Mechanics approach, considering the *quenched entropy*:

$$S=\frac{1}{N}\langle\langle\ln\Omega(\xi^{\mu},\sigma^{\mu})\rangle\rangle \tag{2.2.6}$$

where $\langle\langle\rangle\rangle$ stands for the average with respect to the relevant random quantities.

The calculation of the integral can thus be performed in the thermodinamic limit $N\to\infty$ by the Replica Method outlined in the previous section. One gets:

$$\begin{aligned}\langle\langle\Omega^{n}\rangle\rangle=\int\prod_{a}\frac{d\hat{k}^{a}}{4\pi}\int\prod_{a<b}\frac{dq^{ab}d\hat{q}^{ab}}{2\pi/N}\exp\Bigg(N\Bigg[\frac{i}{2}\sum_{a}\hat{k}_{a}+i\sum_{a<b}q^{ab}\hat{q}^{ab}\\ +G_{S}(\hat{k}^{a},\hat{q}^{ab})+\alpha G_{E}(q^{ab})\Bigg]\Bigg)\end{aligned} \tag{2.2.7}$$

where $G_S(\hat{k}^a,\hat{q}^{ab})$ and $G_E(q^{ab})$ are some functions of the order parameters, called respectively Entropic and Energetic part, in analogy to Statistical Mechanics. Assuming replica symmetry $q^{ab}=q,\hat{q}^{ab}=\hat{q},\hat{k}^{a}=\hat{k}$, and solving the two simple saddle point equation for $\hat{q}$ and $\hat{k}$, one gets a saddle point equation involving q,α,κ. If one then lets $q\to 1$, it is expected that the typical volume of solutions in W space to shrink to a single point, and correspondingly $\alpha\to\alpha_c$, that will be taken as a *critical storage capacity*. By an asymptotic expansion one gets the following equation for α_c:

$$\frac{1}{\alpha_{c}}=\int_{-\infty}^{\kappa}\frac{dt}{\sqrt{2\pi}}e^{-t^{2}/2}(\kappa-t)^{2} \tag{2.2.8}$$

This equation can be solved numerically for $\kappa>0$, yielding the behavior depicted in figure 2.2.1. For $\kappa=0$ one gets a critical capacity $\alpha_c=2$. Note that the same result $\alpha_c=2$ had already been obtained by Cover [**22**] with a completely different approach, namely counting all the possible linear separations (which he called *dichotomies*) of αN points in general positions in an N-dimensional space.

2.2.1. Discrete Perceptron and Replica Symmetry Breaking. As I pointed out in the previous section, one of the main problems in the Replica method is the *stability analysis* of the saddle point: it is a matter of fact that as soon as one switches to the analysis of the discrete perceptron, the RS saddle point is not stable any longer.

Let us begin with the example of an Ising perceptron, obtained with the simple constraint $W_i=\pm 1$. The Replica calculations are quite the same as for the continuous case, with the simple exception of the integration measure replaced by a sum over the W configurations. The result obtained by the RS Ansatz is $\alpha_c=4/\pi\cong 1.27$. This result is quite unrealistic from the informational point of view, since an Ising perceptron with binary synapses can store no more than N bit: one then expects $\alpha_c<1$.

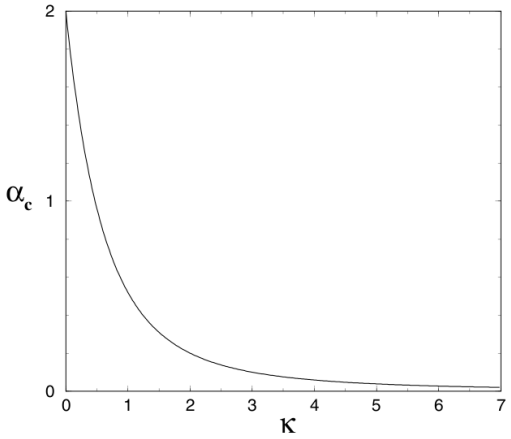

FIGURE 2.2.1. Storage capacity of the spherical perceptron as a function of the stability parameterκ.

The reason for the incorrectness of the result is the instability of the RS Ansatz [**23, 24**]. The next step is then to introduce a 1-RSB Ansatz, with the two parameters q_1 for the intra-cluster overlap, and q_0 for inter-cluster overlap. In the saddle point equation it is then possible to asymptotically expand for $q_1 \to 1$, implying that the volume of a given connected set of solutions shrinks to a single point. It is particularly interesting that in this way one obtains the Zero Entropy condition within the RS Ansatz:

$$s^{RS}(\alpha_c) = 0 \tag{2.2.9}$$

The result obtained is then $\alpha_c \cong 0.83$, in accordance with exact numerical calculations [**25, 26**] and with a later analytical calculation by Gutfreund and Stein [**8**], who treated the general case of local constraints on the synaptic weights as well as biased inputs.

2.2.2. Multi-layer Networks. The perceptron is a rather limited learning machine: it can only realize linear separation of inputs in a certain N-dimensional space. From the logical point of view, its computational abilities are then restricted to linearly separable Boolean functions.

Nevertheless, perceptrons are the basic elements of multi-layer neural networks, that are capable of more general non-linear separations of inputs. I focus on feed-forward neural networks with a single *hidden layer* and a single output unit. From the analytical perspective, it is well know that any function can be approximated by this kind of network with a continuous activation function [**27**]. From the logical perspective, it can be shown that any Boolean function can be implemented in such a network, allowing a binary output [**28**].

Restricting ourselves to tree network structures with a binary output, as depicted in figure 2.2.2, let us define the activation of one of the K hidden units as:

$$\tau_k = \text{sign}\left(\frac{J_k \cdot S}{\sqrt{N}}\right) \tag{2.2.10}$$

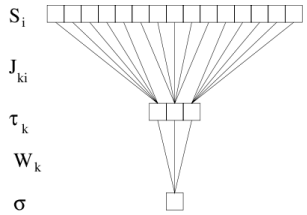

FIGURE 2.2.2. Multi-layer Network with tree structure.

The vector of hidden units activations $\boldsymbol{\tau} = \{\tau_1, ..., \tau_K\}$ is called *internal representation* of a given input S. The output of the network is given by:

$$\sigma = \text{sign}\left(\frac{1}{\sqrt{K}}\sum_{k=1}^{K} W_k \tau_k\right) \tag{2.2.11}$$

It is generally assumed $K \ll N$, so that one can consider that the properties of the network for very large N are not substantially influenced by the variation of the couplings W_k [**29**]. Consequently, the two popular choices for the Boolean function $\sigma = F(\boldsymbol{\tau})$ define two different types of architectures:

- *Parity Machine*

$$F(\tau_1, ..., \tau_K) = \prod_{k=1}^{K} \tau_k \tag{2.2.12}$$

- *Committee Machine*

$$F(\tau_1, ..., \tau_K) = \text{sign}\left(\frac{1}{\sqrt{K}}\sum_{k=1}^{K} \tau_k\right) \tag{2.2.13}$$

2.2.2.1. *Multi-layer networks and statistical bounds.* It is possible to derive some interesting theoretical bounds for the critical capacity of committee machines from purely statistical considerations: the method follows closely the standard approach introduced by Cover [**22**], which consists in counting all the possible binary classifications of pN points in general positions in an N-dimensional space.

Let us call $C(p, N)$ the number of maximal binary classifications of p patterns obtainable with a perceptron. The standard result obtained by Cover was [**22**]:

$$C(p, N) = 2\sum_{i=0}^{N-1} \binom{p-1}{i} \tag{2.2.14}$$

Let us then consider the case of a network with a tree structure, made up of a layer of K units (be it a parity or committee machine). Each of the K sub-perceptrons in the network will be able to realize at most $C(p, N/K)$ classifications. Let us then apply the following known bound [**30**] to the quantity $C(p, N/K)$:

$$\sum_{l=0}^{d} \binom{p}{l} \leq 1.5\frac{p^d}{d!} \leq \left(\frac{ep}{d}\right)^d \tag{2.2.15}$$

The main observation is that the number of internal representations of the network, which stand for all the possible partitions of the p patterns realizable by the K units, has to be higher than the number of possible binary classifications of p patterns. Therefore, using Eq. (2.2.15), one may write:

$$\frac{[C(p,N/K)]^K}{2^p} \leq \exp(N[1+\ln\alpha+\ln K-\alpha\ln 2]) \tag{2.2.16}$$

In the limit of large N one finds an upper bound α_{MD} for the critical capacity α_c, $\alpha_c < \alpha_{MD}$, where α_{MD} is implicitly defined as the value of α at which the ratio in 2.2.16 becomes negligible:

$$1+\ln\alpha_{MD}+\ln K-\alpha_{MD}\ln 2=0 \tag{2.2.17}$$

The above discussion defines the so-called *Mitchinson-Durbin (MD)* bound [**31**], the above formula being valid for a multilayer network with a tree structure. In the limit of a large number of hidden units, one may perform an asymptotic expansion in K, thus obtaining:

$$\alpha_c \leq \alpha_{MD} \sim \frac{\ln K}{\ln 2} \tag{2.2.18}$$

The *MD* upper bound is quite important as a consistency check of the RS calculation, which is always the first step in the Replica approach.

More interestingly, in the particular case of the parity tree an annealed calculation [**32**] of the volume Ω provides exactly the same lower bound $\alpha_c \geq \frac{\ln K}{\ln 2}$, indicating that for large K the asymptotic behavior of α_c is

$$\alpha_c \sim \frac{\ln K}{\ln 2} \tag{2.2.19}$$

2.2.2.2. *Replica theory and multi-layer networks.* The storage problem for multi-layer networks has been attacked with a Replica approach, and what emerged was that Replica Symmetry Breaking is an ubiquitous phenomenon when one deals with such networks, which are capable of constructing mappings from input to output by means of different internal representations.

The RS Replica calculations of the volume Ω on the lines of Gardner's integral (2.2.4) proved very weak in providing the exact results for the critical capacity: expanding the corresponding expressions for α_c in the limit of large K one finds $\alpha_c \sim K^2$ for the parity tree and $\alpha_c \sim \sqrt{K}$ for the committee tree, both expressions violating the Mitchinson-Durbin bound of Eq. (2.2.18). This happens because the RS saddle point is unstable for all values $\alpha > \alpha_{AT}$, $\alpha_{AT} < \alpha_c$. For the parity tree, the results obtained by 1-step RSB seem to be in agreement with the numerical findings, and yields the exact asymptotics (2.2.19).

An interesting approach for the study of the more analitically involved problem of the committee tree is the use of *multifractal methods*, originally introduced in the field of neural networks in the case of the simple perceptron [**33, 34**]. In the following, I will give a very brief discussion of the methods and the main results.

The starting point is the representation of the total volume of compatible vector in the J space as a sum of volumes associated to each internal representation vector:

$$\Omega(\xi^\mu,\sigma^\mu)=\sum_{\boldsymbol{\tau}}\Omega(\boldsymbol{\tau};\xi^\mu,\sigma^\mu) \tag{2.2.20}$$

with

$$\Omega(\boldsymbol{\tau}, \boldsymbol{\xi}^\mu, \sigma^\mu) = \int \prod_{k=1}^{K} d\mu(J_k) \prod_\mu \theta\left(\sigma^\mu F(\tau_1, ..., \tau_K)\right) \prod_{\mu k} \theta\left(\tau_k^\mu \frac{J_k \cdot \xi_k^\mu}{\sqrt{N/K}}\right) \tag{2.2.21}$$

The idea is to measure the fractal dimension of every 'cell' in J space, and then provide the corresponding distribution, namely the so called *multifractal spectrum*. Consider then the quantity

$$k(\boldsymbol{\tau}) = -\frac{1}{N} \ln \Omega(\boldsymbol{\tau}) \tag{2.2.22}$$

and associate to it the *spectrum*

$$c(k) = \frac{1}{N} \ln \mathscr{N}(k) \tag{2.2.23}$$

where $\mathscr{N}(k)$ is the number of cells with a given size k:

$$\mathscr{N}(k) = \sum_{\boldsymbol{\tau}} \delta(k - k(\boldsymbol{\tau})) \tag{2.2.24}$$

The fractal spectrum carries many information about the geometrical structure of the subspace of solutions of a given storage problem. The two most important (and simple) quantities that can be extracted are the following:

(1) the size k_0 of the typical, or most frequent, cell: it is defined as $k_0 = \text{argmax} c(k)$
(2) the size k_1 of the largest cell, dominating the volume under consideration: it can be associated to the maximum of the function $c(k) - k$, as one can write, for large N

$$\Omega = \sum_{\boldsymbol{\tau}} \Omega(\boldsymbol{\tau}) = \int dk \mathscr{N}(k) e^{-Nk} = \int dk e^{(N[c(k)-k])} \tag{2.2.25}$$

The parameter k_0 is very important to study the storage properties of the network: the limit $k \to \infty$ is indeed equivalent to $q \to 1$ in the Gardner approach, because it signals that the typical cell size tends to zero, and provides the critical value α_c. The parameter k_1 is on the other hand associated to the generalization properties of the network.

The computation of the fractal spectrum can be accomplished by a variant of the so called *thermodinamic formalism* for multifractals. Indeed, from the Statistical Mechanics point of view, this approach is simply based on the equivalence of the canonical and micro-canonical ensemble in the thermodynamic limit. Defining then the canonical free energy

$$f(\beta) = -\frac{1}{N\beta} \langle\langle \ln \sum_{\boldsymbol{\tau}} e^{-\beta k(\boldsymbol{\tau})} \rangle\rangle \tag{2.2.26}$$

one obtains the fractal spectrum by a Legendre Transform:

$$c(k) = \min_\beta [\beta k - \beta f(\beta)] \tag{2.2.27}$$

Now, from Eq. (2.2.22) the free energy can be written in the form

$$f(\beta) = -\frac{1}{N\beta}\langle\langle \ln \sum_{\boldsymbol{\tau}} \Omega^{\beta}(\boldsymbol{\tau})\rangle\rangle \tag{2.2.28}$$

The previous expression is suitable for a kind of generalization of the Replica method, with the introduction of two sets of replicas and two exponents, n and r, one of which is not necessairly tending to zero. The calculation of $f(\beta)$ is thus reduced to the Replica integral:

$$\langle\langle \left(\sum_{\boldsymbol{\tau}} \Omega^{\beta}(\boldsymbol{\tau})\right)^{n} \rangle\rangle = \langle\langle \sum_{\boldsymbol{\tau}^a} \int \prod_{k=1}^{K} \prod_{a=1}^{n} \prod_{r=1}^{\beta} d\mu(\boldsymbol{J}_k^{ar}) \prod_{\mu a} \theta\left(\sigma^{\mu} F(\tau_1^{\mu a}, ..., \tau_K^{\mu a})\right) \prod_{\mu k a r} \theta\left(\tau_k^{\mu a} \frac{\boldsymbol{J}_k^{ar} \boldsymbol{\xi}_k^{\mu}}{\sqrt{N/K}}\right) \rangle\rangle \tag{2.2.29}$$

Using this methods, Zecchina and Monasson [**34**] have succeeded in giving the correct expressions for the critical capacity of the parity and committee machine for large K.

The result for the parity machine was the one that I gave previously in Eq. (2.2.19), as obtained by a different method. For the committee machine the asymptotic result is

$$\alpha_c = \frac{16}{\pi}\sqrt{\ln K} \tag{2.2.30}$$

It was further proved [**35**], by a stability analysis of the RS saddle point, that these results are asymptotically correct in the limit $K \gg 1$.

CHAPTER 3

Introduction to Belief Propagation

In the previous section, I dealt with Replica theory and its applications in a seemingly distant field such as neural networks. In this section I will elaborate on Belief Propagation, an important algorithmic implication of the Cavity method, introduced in the context of Spin Glasses, that represents a method complementary to Replicas, with the advantage of making all the assumption on the geometrical structure of the equilibrium states completely explicit [**1**].

Belief Propagation (BP) is a powerful method for computing, at least approximately, the marginal distribution of a variable x_i from a general joint probability distribution $p(x_1, ..., x_N)$ of N variables. This procedure has been discovered independently in different contexts: Statistical Physics (where it is called 'Bethe-Peierls approximation'), Coding Theory (Sum-Product algorithm), Artificial Intelligence (BP). Moreover, one can show that apparently different methods such as the forward-backward algorithm, the Viterbi algorithm, the Kalman filter and the transfer matrix approach in Statistical Physics are all special cases of the BP method.

3.1. BP on tree graphs

Consider a joint probability distribution $p(\underline{x}) \equiv p(x_1, ..., x_N)$, with $x_1, ..., x_N$ taking value in a finite set χ, that can be written as a product of functions ψ_a of the set of variables $\underline{x}_{\partial a}$:

$$p(\underline{x}) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(\underline{x}_{\partial a}) \tag{3.1.1}$$

It is intended that $\underline{x}_{\partial a} \equiv \{x_i | i \in \partial a\}$ and the set ∂a, of size $|\partial a|$, contains all the variables involved in function a. I will call the functions $\psi_a : \chi^{|\partial a|} \to \mathbb{R}$ compatibility functions, or simply constraints.

It can be argued that the mutual dependencies between the variables can be factorized in a non-trivial way, such that distinct variables interact only 'locally'. It is then very helpful to show such inner structure of the probability distribution $p(\underline{x})$ in a graphical way, by means of a so called Factor Graph.

A factor graph is a bipartite graph that contains two types of nodes: N variable nodes, each one associated with a variable x_i (conventionally represented by circles), and M function nodes, each one associated with a function ψ_a (represented by squares). A variable node i is connected by an edge to a function node a if the variable x_i is an argument of $\psi_a(\underline{x}_{\partial a})$. The main feature of a factor graph probability distribution is the so called **Global Markov Property**:

PROPOSITION 1. *Given three disjoint subsets of the variable nodes $A, B, S \subset [N]$, with $\underline{x}_A, \underline{x}_B, \underline{x}_S$ denoting the corresponding sets of variables, if there is no path on the factor graph joining a node of A to a node of B without passing through S then:*

$$P(\underline{x}_A, \underline{x}_B | \underline{x}_S) = P(\underline{x}_A | \underline{x}_S) P(\underline{x}_B | \underline{x}_S)$$

In such a case the variables $\underline{x}_A, \underline{x}_B$ are conditionally independent.

Suppose now that one has to compute the marginal distribution of the variable x_i: the most natural way would be summing over all the configurations of the variables $\{x_j | j \neq i\}$, but from the computational point of view this procedure is extremely time-wasting, due to its exponential complexity $O(\chi^N)$. The Belief Propagation methods stems from a clever way of organizing the sums over the variables, introducing a couple of 'cavity messages' $\nu^{(t)}_{j\to a}$, $\hat{\nu}^{(t)}_{a\to j}$ for each edge (ia) of the graph. The messages are probability distributions defined on the χ space, so:

$$\nu^{(t)}_{j\to a} = \{\nu^{(t)}_{j\to a}(x_j) | x_j \in \chi\}, \qquad \nu^{(t)}_{j\to a}(x_j) \geq 0, \qquad \sum_{x_j} \nu^{(t)}_{j\to a}(x_j) = 1$$

and analogous relations for the $\hat{\nu}^{(t)}_{a\to j}$'s. After initialization as uniform probability distributions on χ, the messages are updated by local computation at any given node of the graph, in accordance to the so called **BP (or sum-product)** update rules:

$$\nu^{(t+1)}_{j\to a}(x_j) \propto \prod_{b\in\partial j\setminus a} \hat{\nu}^{(t)}_{b\to j}(x_j), \tag{3.1.2}$$

$$\hat{\nu}^{(t)}_{a\to j}(x_j) \propto \sum_{\underline{x}_{\partial a\setminus j}} \psi_a(\underline{x}_{\partial a}) \prod_{k\in\partial a\setminus j} \nu^{(t)}_{k\to a}(x_k) \tag{3.1.3}$$

After t iterations, one obtains an estimate of the marginal distribution of the variable i using all its incoming messages:

$$\nu^{(t)}_i(x_i) \propto \prod_{a\in\partial i} \hat{\nu}^{(t-1)}_{a\to i}(x_i) \tag{3.1.4}$$

Moreover, one can interpret all the cavity messages as estimates of marginal distributions in some modified graphs, specifically: $\nu^{(t)}_{j\to a}$ estimates the marginal of x_j in a graph not including the node a; $\hat{\nu}^{(t)}_{a\to j}$ estimates the marginal of x_j in a graph where one has erased all the factor nodes in ∂i except a.

Now, one can search for fixed points of update rules (3.1.2),(3.1.3): the corresponding equations in term of the fixed point messages $\nu^*_{j\to a}, \hat{\nu}^*_{a\to j}$ are called **BP equations**. It is interesting to know whether such equations have a unique solution which Belief Propagation could converge to. The following fundamental theorem states the exactness of BP in factor graphs that have a tree structure:

THEOREM 1. ***B.P. is exact on tree graphs.*** *Consider a tree graphical models with diameter t_*. Then:*

1. *Irrespective of the initial conditions, the BP update converges after at most t_* iterations, that is for any edge (ia), and any $t > t_*$ $\hat{\nu}^{(t)}_{i\to a} = \hat{\nu}^*_{i\to a}, \nu^{(t)}_{i\to a} = \nu^*_{i\to a}$*
2. *The fixed point messages provide the exact marginals, that is for any i and any $t > t_*$ $\nu^{(t)}_i(x_i) = p(x_i)$*

In tree graphs, the fixed point messages can be used to compute the marginal joint distribution of any subset of variables. Consider for example a subset F_R of factors and the corresponding adjacent subset of variables V_R, and denote ∂R the subset of function nodes not in F_R but adjacent to a variable node in V_R. Given the tree structure, one can associate a unique $i \in \partial a \cap V_R$, denoted by $i(a)$ to every factor $a \in \partial R$. It is then easy to realize that:

$$p(\underline{x}_R) = \frac{1}{Z_R} \prod_{a \in F_R} \psi_a(\underline{x}_{\partial a}) \prod_{a \in \partial R} \hat{\nu}^*_{a \to i(a)}(x_{i(a)}) \tag{3.1.5}$$

Applying equation (3.1.5) one can express the marginal distribution of the subset of variables connected to a given factor node a in the following simple way:

$$p(\underline{x}_{\partial a}) = \frac{1}{Z_a} \psi(\underline{x}_{\partial a}) \prod_{j \in \partial a} \nu^*_{j \to a}(x_j) \tag{3.1.6}$$

Expression (3.1.6) is particularly useful to express typical Statistical Mechanics quantities as functions of local messages.

Let us begin with a natural definition of the Internal Energy, writing the compatibility functions as $\psi_a(\underline{x}_{\partial a}) = e^{-\beta E_a(\underline{x}_{\partial a})}$. One defines the internal energy as the expectation value of the total energy (from now on, I take for simplicity $\beta = 1$):

$$U = -\sum_{\underline{x}} p(\underline{x}) \sum_{a=1}^{M} \log \psi_a(\underline{x}_{\partial a}) \tag{3.1.7}$$

By means of Eq. (3.1.6) one gets:

$$U = -\sum_{a=1}^{M} \frac{1}{Z_a} \sum_{\underline{x}_{\partial a}} \psi_a(\underline{x}_{\partial a}) \log \psi_a(\underline{x}_{\partial a}) \prod_{i \in \partial a} \nu^*_{i \to a}(x_j) \tag{3.1.8}$$

In order to express Entropy and, consequently, Free Energy in terms of the messages, I introduce an important result valid in the case of tree graphs:

THEOREM 2. *In a tree graphical model, the joint probability distribution $p(\underline{x})$ of all the variables can be written in terms of the marginals $p_a(\underline{x}_{\partial a})$ and $p_i(x_i)$:*

$$p(\underline{x}) = \prod_{a \in F} p_a(\underline{x}_{\partial a}) \prod_{i \in V} p_i(x_i)^{1-|\partial i|} \tag{3.1.9}$$

From the classical definition of Entropy:

$$H[p] = -\sum_{\underline{x}} p(\underline{x}) \log p(\underline{x})$$

one gets, by Eq. (3.1.9), an expression in term of local quantities:

$$H[p] = -\sum_{a \in F} p_a(\underline{x}_{\partial a}) \log p_a(\underline{x}_{\partial a}) - \sum_{i \in V} (1 - |\partial i|) p_i(x_i) \log p_i(x_i) \tag{3.1.10}$$

Analogously I define the **Bethe free-entropy** as $\Phi = \log Z = H[p] - U[p]$ and obtain:

$$F[p] = -\sum_{a \in F} p_a(\underline{x}_{\partial a}) \log \left\{ \frac{p_a(\underline{x}_{\partial a})}{\psi_a(\underline{x}_{\partial a})} \right\} - \sum_{i \in V} (1 - |\partial i|) p_i(x_i) \log p_i(x_i) \tag{3.1.11}$$

and, in term of local (fixed point) messages:

$$F_*(\underline{\nu}) = \sum_{a\in F} F_a(\underline{\nu}) + \sum_{i\in V} F_i(\underline{\nu}) - \sum_{(ia)\in E} F_{ia}(\underline{\nu}) \tag{3.1.12}$$

where E is the set of links in the graph, and

$$F_a(\underline{\nu}) = \log\left[\sum_{\underline{x}_{\partial a}} \psi_a(\underline{x}_{\partial a}) \prod_{j\in\partial a} \nu_{j\to a}(x_j)\right], \qquad F_i(\underline{\nu}) = \log\left[\sum_{x_j} \prod_{b\in\partial j} \hat{\nu}_{b\to j}(x_j)\right],$$

$$F_{ia}(\underline{\nu}) = \log\left[\sum_{x_j} \nu_{j\to a}(x_j)\hat{\nu}_{a\to j}(x_j)\right]$$

The usefulness of Belief Propagation is not limited to the computation of marginals and free-entropy (and so partition function): it can be used, for example, to sample from a given probability distribution [**2**], or to find the configuration $\underline{x}$ that maximizes the probability $p(\underline{x})$, by a decimation procedure.

3.2. BP on general factor graphs

There is no general theory about Belief Propagation on factor graphs with loops. In principle, one can use the formulas (3.1.8),(3.1.10),(3.1.12) as definitions for the BP estimates of the corresponding quantities. One can, though, rely on a variational approach connecting **Bethe free-entropy** to BP fixed points in general graphs. Indeed, free-entropy can be regarded as a function defined in the space of the messages or, equivalently, on the space of all possible **locally consistent marginals**, also called **beliefs**. More specifically, let us consider the distributions $b_i(x_i) \geq 0, b_a(\underline{x}_{\partial a}) \geq 0$ for each variable and function node in a graph, satisfying the normalization conditions:

$$\sum_{x_i} b_i(x_i) = 1 \quad \forall i \in V, \qquad \sum_{\underline{x}_{\partial a}} b_a(\underline{x}_{\partial a}) = 1 \quad \forall a \in F$$

If they satisfy the following normalization conditions:

$$\sum_{\underline{x}_{\partial a\setminus i}} b_a(\underline{x}_{\partial a}) = b_i(x_i), \quad \forall a \in F, \forall i \in \partial a \tag{3.2.1}$$

they are a set of locally consistent marginals. One then build the **Bethe free-energy** of this set:

$$F[\underline{b}] = -\sum_{a\in F} b_a(\underline{x}_{\partial a}) \log\left\{\frac{b_a(\underline{x}_{\partial a})}{\psi_a(\underline{x}_{\partial a})}\right\} - \sum_{i\in V}(1-|\partial i|)b_i(x_i)\log b_i(x_i) \tag{3.2.2}$$

The following theorem marks an important connection between free-entropy and BP equations, analogous to the variational justification for the use of the standard mean field approximation [**36**].

THEOREM 3. *Assume $\psi(\underline{x}_{\partial a}) > 0$ for each $a \in F$ and for all $\underline{x}_{\partial a} \in \chi^{|\partial a|}$. Then the stationary points of the Bethe free-entropy $F[\underline{b}]$ are in one-to-one correspondence with the fixed points of BP.*

This result directly implies the existence of at least one BP fixed point in a factor graph with a general structure. Moreover, one could argue that Belief Propagation represents a kind of generalization of the mean field approximation, in which local binary interactions between variables are taken into account.

A general theory for the correctness of the BP equations is unfortunately lacking. Correctness has been established for arbitrary topologies in a few particular cases in the zero temperature limit [**37, 38, 39**] and in the case of gaussian potentials [**40**]. Regarding the problem at non-zero temperature, approximation bounds and correctness of the marginals and the free energy can be ensured (at least in the infinite volume limit) for locally tree-like factor graphs with potentials leading to map contractiveness of the BP equations or more general correlation decay conditions like De Dobrushin's, see e.g. Ref. [**41**].

CHAPTER 4

Perceptron Learning and subdominant states

In this chapter I will go through several aspects of the learning problem in perceptrons with discrete synapses. There are a number of reasons why this particular case is interesting: recent experimental evidence [**42, 43**] suggests that synapses have a limited precision, and are able to store up to a few bits each (between 1 and 5); moreover, that of discrete (binary in particular) synaptic weights is the natural scenario for neuromorphic applications, that aim at a hardware implementation of classification devices, in principle able to modify their structure locally and autonomously.

Even in its simplest formulation, the problem of learning with discrete synapses is known to be intractable in the worst-case [**44**]. In the typical case, the classical Statistical Mechanics description of the model shows that it is dominated in the large N limit by an exponential number (in N) of local minima [**24, 45, 46, 47**], which easily trap standard search strategies based on energy minimization, e.g. Monte Carlo [**48, 49**] (a situation typical of spin glass phase, which is common to many hard random optimization problems [**50, 2, 51**]). This description seems to be in contrast with a series of algorithmic results [**52, 53, 54**] which show that nearly optimal performance can be achieved by heuristic modifications of the original Belief Propagation approach: the striking ability of simple learning algorithms raises the issue of what are the actual features of the particular solutions these methods are able to find, and to what extent they correspond to the typical solutions which can be found in the standard equilibrium Replica calculations.

This chapter is organized as follows: the first section will be devoted to the formal definition of the problem with a survey of theoretical results, together with an introduction to the BP-derived heuristic algorithms; in section 4.2, I will describe the problem of clustered solutions that motivated the investigation of the relevance of subdominant states in the present problem. Section 4.3 is a very brief account of a possibile extension of the learning algorithm for binary weights to a network with one hidden layer, in the context of a multi-label classification task.

4.1. Learning in Perceptron

As I pointed out in section 2.2, the perceptron implements a binary classification of some patterns ξ's with a separating hyperplane orthogonal to a vector W, whose components are called weights, or synapses, in a very rough analogy to the integration and thresholding operations that happen in a post-synaptic neuron with respect to its presynaptic inputs. In particular, a binary perceptron maps vectors of N inputs $\xi \in \{-1,1\}^N$ to binary outputs $\tau(W,\xi) = \mathrm{sign}(W \cdot \xi)$ via a vector of binary (*Ising*) synaptic weights $W \in \{-1,1\}^N$. Let us now consider the case of a number of patterns that is proportional to dimension N, namely the perceptron is given αN input patterns ξ^μ with $\mu \in \{1,\dots,\alpha N\}$, the ξ_i^μ being random unbiased i.i.d. variables, together with their corresponding desired outputs $\sigma^\mu \in \{-1,1\}^{\alpha N}$. The learning problem consists in finding a vector W which correctly classifies the inputs ξ^μ. More formally, let us define an indicator function $\mathbb{X}_\xi(W) = \prod_{\mu=1}^{\alpha N} \Theta(\sigma^\mu \tau(W,\xi^\mu))$,

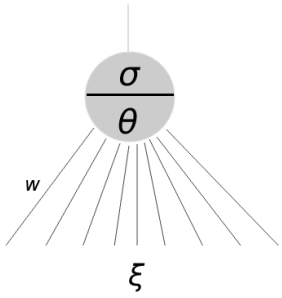

FIGURE 4.1.1. The binary Perceptron classifies inputs ξ's via a projection on a weight vector W, producing the binary output σ; θ is an additional threshold parameter that has to be defined in case the binary synaptic weights are chosen in the set $\{0, 1\}$.

where $\Theta(x)$ is the Heaviside step function: the learning problem consists in finding W such that $\tau(W, \xi^\mu) = \sigma^\mu$ for all μ, i.e. such that $\mathbb{X}_\xi(W) = 1$.

The two main instantiations of the learning problem which have been thoroughly studied are the *classification* case, in which the outputs σ^μ are chosen independently at random, and the *generalization* (or *teacher-student*) case, in which a "student" perceptron W has to learn a classification rule that has been defined by another randomly chosen perceptron $W^{\mathscr{T}}$, which sets the outputs σ^μ. As I briefly discussed in section 2.2, the classical Gardner approach consists in carrying a standard zero-temperature equilibrium analysis of the model with the construction of a probability measure $p(W) = \mathbb{X}_\xi(W)/Z_{eq}$, with $Z_{eq} = \sum_{\{W\}} \mathbb{X}_\xi(W)$ the partition function; the typical case is described by taking the quenched average $\langle \log(Z_{eq}) \rangle_\xi$ over the realizations of the patterns. In the large N limit, both the classification and the generalization case show a sharp transition at a well defined value of the parameter α, called the *capacity*. In particular the typical classification problem has a solution with probability 1 in the limit of large N up to $\alpha_c = 0.833$ [**24**], after which the probability of finding a solution drops to zero. In the teacher-student scenario, the problem has exponentially many solutions up to $\alpha_{TS} = 1.245$, after which there is a first-order transition and the only solution (the one that minimizes the free energy of the system) is the teacher $W^{\mathscr{T}}$ itself [**45, 14**]. Since the teacher is known in advance, one generally measures the generalization error rate $p_e = \frac{1}{\pi} \arccos\left(\frac{1}{N} W \cdot W^{\mathscr{T}}\right)$, which is the probability that the student is able to correctly classify a previously unseen input $\xi^\star$, i.e. $\tau(W, \xi^\star) = \tau\left(W^{\mathscr{T}}, \xi^\star\right)$. In the following sections, I will describe the Belief Propagation machinery that can be used not only to compute the quantity $\log(Z_{eq})$ in the single instance of a learning problem, but also, more importantly, to search for a solution W.

4.1.1. Learning by Belief Propagation. Suppose the network is given a set of $p = \alpha N$ input-output associations $\{\xi^\mu\} \to \{\sigma^\mu\}$. One will associate a factor graph to the learning problem set by the distribution $p(W) \propto \mathbb{X}_\xi(W)$, which is composed of p copies of the basic structure depicted in Fig. 4.1.2, each for one pattern to learn. Just for the sake of the argument, I introduced an additional variable node s with an intermediate function node that implement the sum constraint $\mathbb{I}\left(\sum_j \xi_j^\mu W_j = s^\mu\right)$, where $\mathbb{I}$ is the *characteristic function*, whose output value is 1 if the constraint is satisfied and 0 otherwise. Let us now consider a sum function node in the graph, as depicted in Fig. (4.1.3): updating the cavity messages associated to its links is computationally very expensive, since one needs to perform a trace that is exponential in N. The update equations for such a node are the

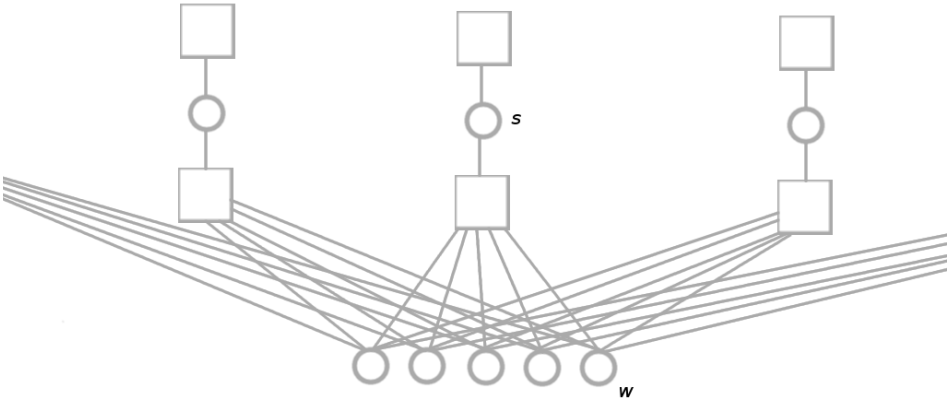

FIGURE 4.1.2. The factor graph of the perceptron learning problem.

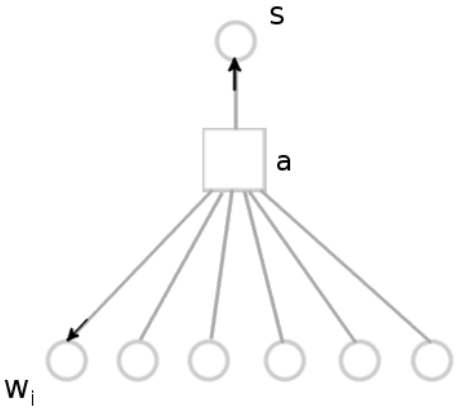

FIGURE 4.1.3. Pictorial representation of the messages to and from a sum node (update equations (4.1.1), (4.1.2)) .

following:

$$\hat{\nu}_{a\to s}^{(t+1)}(s) \propto \sum_{\{w_j\}} \mathbb{I}\left(\sum_j \xi_j W_j = s\right) \prod_j \nu_{j\to a}^{(t)}(W_j) \tag{4.1.1}$$

$$\hat{\nu}_{a\to i}^{(t+1)}(W_i) \propto \sum_s \nu_{s\to a}^{(t)}(s) \sum_{\{W_j|j\neq i\}} \mathbb{I}\left(\sum_{\{j|j\neq i\}} \xi_j W_j + \xi_i W_i = s\right) \prod_{j\neq i} \nu_{j\to a}^{(t)}(W_j) \tag{4.1.2}$$

The sum on the configurations of all the variables W_j appearing in (4.1.1),(4.1.2), which can be put in the form of a series of $N-1$ convolutions between the messages $\nu_{j\to a}$, makes the update a process with exponential complexity $O(2^N)$, depending on which post-synaptic potential is under consideration. Moreover, in a network with p patterns, there are αN sum nodes, and this makes the update step of the algorithm very slow even for small values of N.

It is then useful to introduce a gaussian approximation, that stems from an application of the Central Limit Theorem to the sum of the quantities $\xi_j W_j$ with the indipendent distributions of cavity messages $\nu_{j\to a}$. In particular, one obtains:

$$\hat{\nu}_{a\to s}^{(t+1)}(s) \approx \int dz G_{(\mu_a,\sigma_a^2)}(z)\, \mathbb{I}(z=s) \tag{4.1.3}$$

$$\hat{\nu}_{a\to i}^{(t+1)}(W_i) \approx \sum_s \nu_{s\to a}^{(t)}(s) \int dz G_{\left(\mu_{a\setminus i},\sigma_{a\setminus i}^2\right)}(z)\, \mathbb{I}(z+\xi_i W_i = s) \tag{4.1.4}$$

In the previous approximated equations, the sum over the configurations has been replaced by an integral over a gaussian distribution G whose mean and variance are determined by the corresponding messages appearing in the BP updates:

$$\mu_a = \sum_j \xi_j \mathbb{E}_{\nu_{j\to a}^{(t)}}(W_j), \qquad \sigma_a^2 = \sum_j \mathbb{VAR}_{\nu_{j\to a}^{(t)}}(W_j) \tag{4.1.5}$$

$$\mu_{a\setminus i} = \sum_{\{j|j\neq i\}} \xi_j \mathbb{E}_{\nu_{j\to a}^{(t)}}(W_j), \qquad \sigma_{a\setminus i}^2 = \sum_{\{j|j\neq i\}} \mathbb{VAR}_{\nu_{j\to a}^{(t)}}(W_j) \tag{4.1.6}$$

As I said in Chapter 3, BP equations will eventually provide the marginals $p(W_i)$ over the original distribution, together with an estimate for the entropy or other average quantities of interest. Here, I am mostly interested in the possibility of turning BP into an actual solver. In what follows, I will use the term capacity also to define the maximum number of input-output associations that can be successfully stored in a perceptron with a given algorithm in the limit of large N. Up to now, there are only a handful of heuristic algorithms that are able to solve the classification problem and achieve a non-zero capacity in the limit of large N in a sub-exponential running time: reinforced Belief Propagation (R-BP) [**52**], reinforced Max-Sum (R-MS) [**55**], SBPI [**53**] and CP+R [**54**]. I will give a brief account for each of them in the following section. In the classification case, they achieve capacities between $\alpha \simeq 0.69$ and $\alpha \simeq 0.75$. They all share the property of being local and distributed, and have typical solving times which scale almost linearly with the size of the input. SBPI and CP+R additionally have extremely simple requirements (only employing finite discrete quantities and simple, local and on-line update schemes), making them appealing for practical purposes and reasonably plausible candidates for biological implementations. A qualitatively similar scenario holds in the generalization case, where all these algorithms perform well except in a finite window $1 \lesssim \alpha \lesssim 1.5$ around α_{TS}.

4.1.2. Brief description of the heuristic algorithms. The R-BP algorithm is a variant of the standard Belief Propagation (BP) algorithm. The BP algorithm can be turned into a heuristic solver by adding a reinforcement term: this is a time-dependent external field which tends to progressively polarize the probability distributions on a particular configuration, based on the approximate marginals computed at preceeding steps of the iteration of the BP equations. The reinforcement can thus be seen as a "soft decimation" process, in which the variables are progressively and collectively fixed until they collapse onto a single configuration. This method seems to have an algorithmic capacity of at least $\alpha \simeq 0.74$.

The R-MS algorithm is analogous to the R-BP algorithm, using Max-Sum (MS) as the underlying algorithm rather then BP. The MS algorithm can be derived as a particular zero-temperature limit of the BP equations, or it can be seen as a heuristic extension of the dynamic programming approach to loopy graphs. The reinforcement term acts in the same way as previously described for R-BP. The resulting characteristics of R-MS are very similar to those of BP; extensive numerical tests give a capacity of about $\alpha \simeq 0.75$.

The SBPI algorithm was derived as a crude simplification of the R-BP algorithm, the underlying idea being that of stripping R-BP of all features which would be completely unrealistic in a biological context. This resulted in an on-line algorithm, in which patterns are presented one at a time, and in which only information locally available to the synapses is used in the synaptic update rule. Furthermore, the algorithm only uses a finite number of discrete internal states in each synapse, and is remarkably robust to noise and degradation. Rather surprisingly, despite the drastic simplifications, the critical capacity of this algorithm is only slightly reduced with respect to the original R-BP algorithm, and was measured at about $\alpha \simeq 0.69$.

The CP+R algorithm was derived as a further simplification of the SBPI algorithm. It is equivalent to the former in the context of the on-line generalization task, but requires some minor modifications in the classification context. Its main difference from SBPI is that it substitutes an update rule which was triggered by near-threshold events in SBPI with a generalized, stochastic, unsupervised synaptic reinforcement process (the rate of application of this mechanism needs to be calibrated for optimal results). The kind of reinforcement mentioned here is rather different from the reinforcement term of R-BP or R-MS. The capacity of the CP+R algorithm can be made equal to that of SBPI, $\alpha \simeq 0.69$.

4.2. Large deviation analysis

The striking effectiveness of the simple learning protocols that have been described in the previous section is in contrast with the standard Replica analysis, in particular with a picture of a glassy energy landscape in which solutions are isolated. The geometrical structure of the space of solutions has been recently studied by Huang and Kabashima [**49**] by means of the Franz-Parisi potential [**56**], a useful method in Spin Glass physics that allows to study the role of metastable states. The authors performed a Replica calculation of the following expression:

$$\mathscr{F}_{FP}(\beta,\beta',\gamma) = \left\langle \frac{\sum_{\tilde{W}} e^{-\beta' \sum_\mu \theta\left(-\sigma^\mu \sum_j \tilde{W}_j \xi_j^\mu\right)}}{Z} \log\left(\sum_W e^{-\beta \sum_\mu \theta\left(-\sigma^\mu \sum_j W_j \xi_j^\mu\right) + \gamma \sum_j \tilde{W}_j W_j} \right) \right\rangle \tag{4.2.1}$$

Introducing the energy function $E(W,\{\xi\}) = \sum_\mu \theta\left(-\sigma^\mu \sum_j W_j \xi_j^\mu\right)$, one recognizes in expression (4.2.1) a 'constrained' free energy of a system with Hamiltonian $E(W,\{\xi\})$ at the inverse temperature β which is coupled to another system $E(\tilde{W},\{\xi\})$ at inverse temperature β' via a field $\gamma \sum_j \tilde{W}_j W_j$, controlled by the parameter γ. The main idea is that the second system acts on the first one as a quenched disorder, so that γ effectively controls the distance between the reference $\tilde{W}$ and the

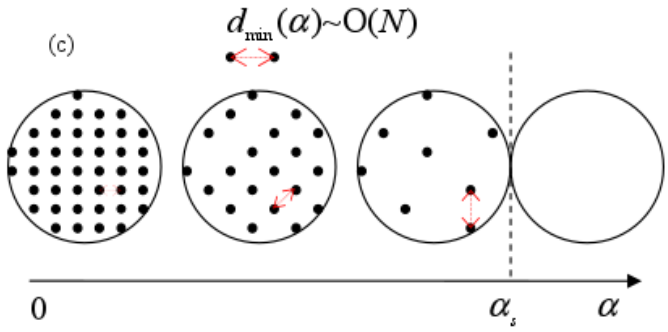

FIGURE 4.2.1. Pictorial depiction of the structure of the space of solutions in the storage problem based on the results of the Franz-Parisi analysis in the typical case. The storage capacity is indicated by $\alpha_s \approx 0.833$. Figure is taken from Ref. [**49**].

neighboring configurations W. In the zero temperature limit $\beta = \beta' \to \infty$, the quantity $\mathscr{F}_{FP}(\gamma)$ is dominated by the ground states of the systems, so that one studies the local structure of the space of solutions.

The authors computed the Franz-Parisi potential via a Legendre transform $\mathscr{S}_{FP}(S) = \mathscr{F}_{FP}(\gamma) - \gamma S$, with $S = \frac{\sum_j \tilde{W}_j W_j}{N}$ the overlap between the configurations of the two systems. The quantity $\mathscr{S}_{FP}(S)$ is logarithm of the number of solutions W which are at a certain distance $d = (1-S)/2$ from a reference solution $\tilde{W}$. The main result of the calculation was that there exists a minimal extensive distance between the zero energy configurations of the two systems, meaning that a typical equilibrium solution $\tilde{W}$ has a sub-exponential number of neighboring solutions W, in a way that is depicted in Fig. (4.2.1). Nevertheless, when one investigates this issue numerically, several evidence is found that the solutions found by the algorithms are typically not isolated; rather, they belong (with high probability at large N) to large connected clusters of solutions:

(1) if one starts a random walk process in a given solution $\tilde{W}$, constraining neighboring configurations to be solutions, one can reach distances of order N from the starting point;
(2) the number of solutions at a distance of order N from $\tilde{W}$ grows exponentially with N (this can be estimated from the analysis of the recurrence relations on the average growth factor of the number of solutions at varying distances, and using the random walk process for sampling the local properties relevant to those relations) .

Belief Propagation can be used to investigate the local structure of the space of solution as well: once a solution $\tilde{W}$ is found with, for instance, CP+R, it can be coupled to the BP variable nodes W as an external field (I will be more precise in the following chapter, and provide a different but equivalent form of equations 4.1.3 -4.1.4 in the Appendix A.3). This is the single instance BP equivalent of the Franz-Parisi potential [**56**], in that it allows to analyze the neighboring solutions with respect to a reference$\tilde{W}$.

What I found in the classification case was that the results do not match the predictions of the equilibrium analysis in Ref. [**49**]: Fig. 4.2.2 shows, in fact, that there is a finite entropy of solutions up to indefinitely high values of the Franz-Parisi coupling γ, corresponding to arbitrarily small distances d of W from the reference $\tilde{W}$.

The same analysis in the teacher student case yielded very similar results: the teacher device is isolated and indistinguishable from all other typical solutions except for the generalization error, so that the estimates obtained from BP are consistent with the analytical calculation when using the teacher as a reference point, but not when using a solution provided by a heuristic solver (see inset in Fig. 4.2.2). Very interestingly, the generalization error for solutions found algorithmically is lower than what would be expected for a typical solution (see Fig. 4.2.4).

The first tentative approach was then to extend the equilibrium analysis of Ref. [**49**] to the teacher-student scenario, where we found that typical solutions are isolated for all values of α even when adding a non-zero stability constraint, as in the classification case. These results indicate that calculations performed at thermodynamic equilibrium are effectively blind to the solutions found by the heuristic algorithms.

We were led to introduce a large-deviation description of this regime, which — according to the numerical evidence — is characterized by regions with a high density of solutions. What we did was then to modify the statistical weight of the individual solutions in a way that favors the ones which are surrounded by a large number of other solutions. Therefore, we studied the following large-deviation free energy density function:

$$\mathscr{F}(d,y) = -\frac{1}{Ny}\log\left(\sum_{\{\tilde{W}\}} \mathbb{X}_\xi\left(\tilde{W}\right)\mathcal{N}\left(\tilde{W},d\right)^y\right) \tag{4.2.2}$$

where $\mathcal{N}\left(\tilde{W},d\right) = \sum_{\{W\}} \mathbb{X}_\xi(W)\,\delta\left(W\cdot\tilde{W}, N(1-2d)\right)$ counts the number of solutions W at normalized Hamming distance d from a reference solution $\tilde{W}$ (δ is the Kronecker delta symbol), and y has the role of an inverse temperature. This free energy describes a system in which each configuration $\tilde{W}$ is constrained to be a solution, and has a formal energy density $\mathscr{E}\left(\tilde{W}\right) = -\frac{1}{N}\log\mathcal{N}\left(\tilde{W},d\right)$ which favors configurations surrounded by an exponential number of other solutions, with y controlling the amount of reweighting. The regions of highest local density are then described in the regime of large y and small d.

The relevant quantities are computed through the usual statistical physics tools. Of particular importance is the entropy density of the surrounding solutions, the *local entropy*:

$$\mathscr{S}_I(d,y) = -\left\langle \mathscr{E}\left(\tilde{W}\right)\right\rangle_{\xi,\tilde{W}} = \frac{1}{N}\left\langle \log\mathcal{N}\left(\tilde{W},d\right)\right\rangle_{\xi,\tilde{W}}. \tag{4.2.3}$$

which is simply given by $\mathscr{S}_I(d,y) = \partial_y\left(y\mathscr{F}(d,y)\right)$. The existence of a dense and exponentially large cluster of solutions will be signaled by a positive entropy $\mathscr{S}_I(d,y) > 0$ in a neighborhood of $d = 0$. Another important quantity is the *external entropy*, i.e. the entropy of the reference solutions $\mathscr{S}_E(d,y) = -y\left(\mathscr{F}(d,y) + \mathscr{S}_I(d,y)\right)$, which must also be non-negative. The special case $y = 1$ is essentially equivalent to the computation of Ref. [**46**]; $\mathscr{S}_I(d,y)$ reduces to the computation *à la* Franz-Parisi of Ref. [**49**] in the limit $y \to 0$.

We computed the action $\phi = -y\mathscr{F}$ corresponding to the free energy $\mathscr{F}$ of Eq. 4.2.2 by the Replica method, in the Replica symmetric Ansatz. The resulting expression, in the generalization case, is:

$$\begin{aligned}(4.2.4)\quad &= -\frac{1}{2}(1-\tilde{q})\,\hat{\tilde{q}} - \frac{y}{2}(1-q_1)\,\hat{q}_1 - \frac{y^2}{2}(q_1\hat{q}_1 - q_0\hat{q}_0) + y\tilde{S}\hat{\tilde{S}} - yS\hat{S} - \tilde{R}\hat{\tilde{R}} - yR\hat{R} + \mathcal{G}_S + \alpha\mathcal{G}_E\end{aligned}$$

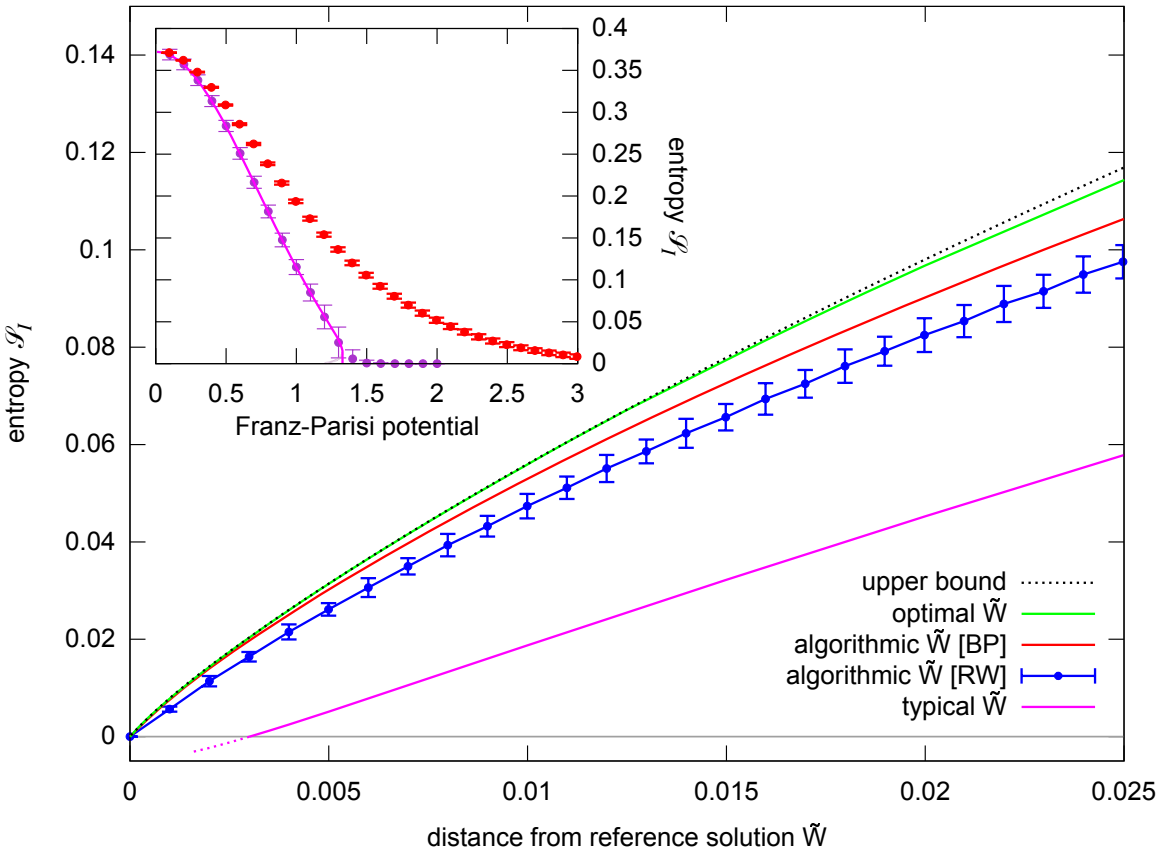

FIGURE 4.2.2. (Color online) **Numerical evidence of the existence of clusters of solutions**. Entropy at a given distance from a reference solution $\tilde{W}$, in the classification case at $\alpha = 0.4$. From bottom to top: (magenta) theoretical prediction for a typical $\tilde{W}$; (blue) numerical estimate based on a random walks on connected solutions starting from one provided by SBPI, with $N = 1001$; (red) estimate from Belief Propagation using a solution from SBPI, with $N = 10001$; (green) theoretical curve for the optimal $\tilde{W}$ as computed from eq. (4.2.2); (dotted black) upper bound ($\alpha = 0$ case, all configurations are solutions). The random-walk points underestimate the number of solutions since they only consider single-flip-connected clusters; the BP curve is lower than the optimal because in the latter $\tilde{W}$ is optimized as a function of the distance, while in the former it is fixed. *Inset*: comparison between a typical solution and one found with SBPI, in the teacher-student case at $\alpha = 0.5$ with $N = 1001$. Larger potentials correspond to smaller distances. Top points (red): SBPI reference solution, entropy computed by BP; bottom curve (magenta): theoretical prediction for a typical solution; bottom points (purple): BP results using the teacher as reference.

where we used the overlap $S = 1 - 2d$ as a control parameter instead of d, and

$$\mathcal{G}_S = \int D\tilde{z} \int Dz_0 \log \sum_{\tilde{W}=\pm 1} \exp\left(\tilde{W}\tilde{A}\left(\tilde{z}, z_0\right)\right) \int Dz_1 \left(2\cosh\left(A\left(z_0, z_1, \tilde{W}\right)\right)\right)^y$$

$$\mathcal{G}_E = 2\int D\tilde{z} \int Dz_0\, H\left(\eta\left(\tilde{z}, z_0\right)\right) \log \int Dz_1 H\left(\tilde{C}\left(\tilde{z}, z_0, z_1\right)\right) H(C\left(z_0, z_1\right)^y$$

$$\begin{aligned}
\tilde{A}\left(\tilde{z}, z_0\right) &= \tilde{z}\sqrt{\hat{\tilde{q}} - \frac{\hat{\tilde{S}}^2}{\hat{q}_0}} + z_0 \frac{\hat{\tilde{S}}}{\sqrt{\hat{q}_0}} + \hat{\tilde{R}} \\
A\left(z_0, z_1, \tilde{W}\right) &= z_1\sqrt{\hat{q}_1 - \hat{q}_0} + z_0\sqrt{\hat{q}_0} + \hat{R} + \tilde{W}\left(\hat{S} - \hat{\tilde{S}}\right) \\
\eta\left(\tilde{z}, z_0\right) &= \frac{wRz_0 + \left(q_0\tilde{R} - R\tilde{S}\right)\tilde{z}}{\sqrt{q_0}\sqrt{w^2 - \left(\tilde{q}R^2 + q_0\tilde{R}^2 - 2R\tilde{R}\tilde{S}\right)}} \\
w &= \sqrt{\tilde{q}q_0 - \tilde{S}^2} \\
C\left(z_0, z_1\right) &= \frac{z_1\sqrt{q_1 - q_0} + z_0\sqrt{q_0}}{\sqrt{1 - q_1}} \\
\tilde{C}\left(\tilde{z}, z_0, z_1\right) &= \frac{\left(S - \tilde{S}\right)z_1 + \sqrt{\frac{q_1 - q_0}{q_0}}\left(\tilde{S}z_0 + w\tilde{z}\right)}{\sqrt{\left(q_1 - q_0\right)\left(1 - \tilde{q}\right) - \left(S - \tilde{S}\right)^2}}
\end{aligned}$$

The order parameters $\tilde{q}$, q_1, q_0, $\tilde{S}$, R, $\tilde{R}$ and their conjugates ($\hat{\tilde{q}}$, $\hat{q}_1$ etc. and $\hat{\tilde{S}}$) must be determined from the saddle point equations, i.e. by setting to zero the derivative of $\phi\left(S, y\right)$ with respect to each parameter. This yields a system of 13 coupled equations, with α, y and S as control parameters. These equations were solved by iteration.

The physical interpretation of the order parameters is as follows (here, the overlap between two configurations X and Y is defined as $\frac{1}{N}\left(X \cdot Y\right)$):

- $\tilde{q}$**:** overlap between two different reference solutions $\tilde{W}$
- q_1**:** overlap between two solutions W referred to the same $\tilde{W}$
- q_0**:** overlap between two solutions W referred to two different $\tilde{W}$
- S**:** overlap between a solution W and its reference solution $\tilde{W}$
- $\tilde{S}$**:** overlap between a solution W and an unrelated reference solution $\tilde{W}$
- R**:** overlap between a solution W and the teacher $W^{\mathcal{T}}$
- $\tilde{R}$**:** overlap between a reference solution $\tilde{W}$ and the teacher $W^{\mathcal{T}}$

Therefore, $\tilde{R}$ can be used to compute the typical generalization error of reference solutions $\tilde{W}$, as $\frac{1}{\pi}\arccos\left(\tilde{R}\right)$. An analogous relation yields the generalization error of the solutions W as a function of R. It is also worth noting that $\tilde{q} < 1$ implies that the number of reference solutions $\tilde{W}$ is larger than 1. By setting to zero the order parameters R, $\tilde{R}$ and their conjugates, and thus reducing the system of equations to the remaining 9 saddle point conditions, we obtain the classification scenario.

It can be noted that, although this solution was called Replica symmetric, the structure is highly reminiscent of a 1-RSB solution. Indeed, it can be shown that, if the constraints on the configurations $\tilde{W}$ are removed, and one solves for $\tilde{S} = 0$ rather then fixing S, one obtains exactly the standard 1-RSB

equations for the perceptron of [**24**] at zero temperature, with y taking the role of the Parisi parameter m. However, 1-RSB solution of the standard equations shows no hint of the dense regions, even if one relaxes the requirement $0 \leq m \leq 1$ of [**24**]. This shows that the constraint on the distance is crucial to explore these subdominant regions.

From Eq. (4.2.4) one can compute the internal and external entropies, as:

$$\mathscr{S}_I(S,y) = \frac{\partial \phi}{\partial y}(S,y) \tag{4.2.5}$$

$$\mathscr{S}_E(S,y) = \phi(S,y) - y\frac{\partial \phi}{\partial y}(S,y) \tag{4.2.6}$$

What one finds is that, for all values of α and d, there is a value of y beyond which $\mathscr{S}_E(d,y) < 0$, which signals a problem with the RS assumption. Therefore, we sought numerically for each α and S (and correspondingly d) the value $y^\star = y^\star(\alpha, d)$ at which $\mathscr{S}_E(d, y^\star) = 0$, i.e. the highest value of y for which the RS analytical results are consistent. Therefore, in all our results, the typical number of reference solutions $\tilde{W}$ was sub-exponential in N; however, we found that in all cases $\tilde{q} < 1$, which implies that the solutions $\tilde{W}$ are not unique.

Using the value of the temperature at which the (external) entropy vanishes is sufficient in this case to derive results which are geometrically valid across most values of the control parameters α and S. There are two exceptions to this observation, both occurring at high values of α and in specific regions of the parameter S. Let us indicate with $[S_L, S_R]$ these regions, with $0 < S_L < S_R < 1$. The most obvious kind of problem occurs occurs at $\alpha \gtrsim 0.79$, where $\mathscr{S}_I(S,y) < 0$ for $S \in [S_L, S_R]$. Another type of transition occurs between $\alpha \simeq 0.77$ and $\alpha \simeq 0.79$, where the $\frac{\partial}{\partial S}\mathscr{S}_I(S,y) \geq 0$ in $[S_L, S_R]$. A closer inspection of the order parameters reveals that, $q_1 \geq S$ for $S \in [S_L, S_R]$. The transition points S_L and S_R at which $q_1 = S$ are manifestly unphysical, because in that case any of the solutions W (which are exponential in number, since $\mathscr{S}_I > 0$) could play the role of the reference solution $\tilde{W}$, and yet the number of $\tilde{W}$ should be sub-exponential, because $\mathscr{S}_E = 0$. This is a contradiction: those regions are then inadequately described within the RS Ansatz.

As for the parts of the curves which are outside these problematic regions, the results obtained under the RS assumption are reasonable, and in very good agreement with the numerical evidence. In order to assess whether the RS equations are stable, further steps of RSB would be needed; unfortunately, this would multiply the number of order parameters (and thus enlarge the system of equations) and the number of nested integrals required for each of these equations. In conclusion, the results described here suggest that the general picture is well captured by the RS assumption with the zero external entropy requirement, and that quantitative adjustments due to further levels of RSB would likely be small, and limited to the intermediate regions of S.

The solution to the system of equations stemming from the RS saddle point produces qualitatively very similar results for both the classification (with $\alpha < \alpha_c$) and the generalization (with $\alpha < \alpha_{TS}$) case. Some results in the classification case are sketched in Fig. 4.2.3. The main facts are described in what follows:

(1) For all $\alpha < \alpha_c$, there is a neighborhood of $d = 0$ where $\mathscr{S}_I(d) > 0$, implying the existence of extensive clusters of solutions. Furthermore, for all α, the curves for $\mathscr{S}_I(d)$ are all approximately equal around $d = 0$; in particular, they all approximate the case for $\alpha = 0$ where all points are solutions. This implies that the clusters of solutions are extremely dense at their core. The size of this dense region shrinks with α and vanishes at α_c.
(2) For large distances, as expected, $\mathscr{S}_I(d)$ collapses with a second-order transition onto the equilibrium entropy, i.e. this regime is dominated by the typical solutions.

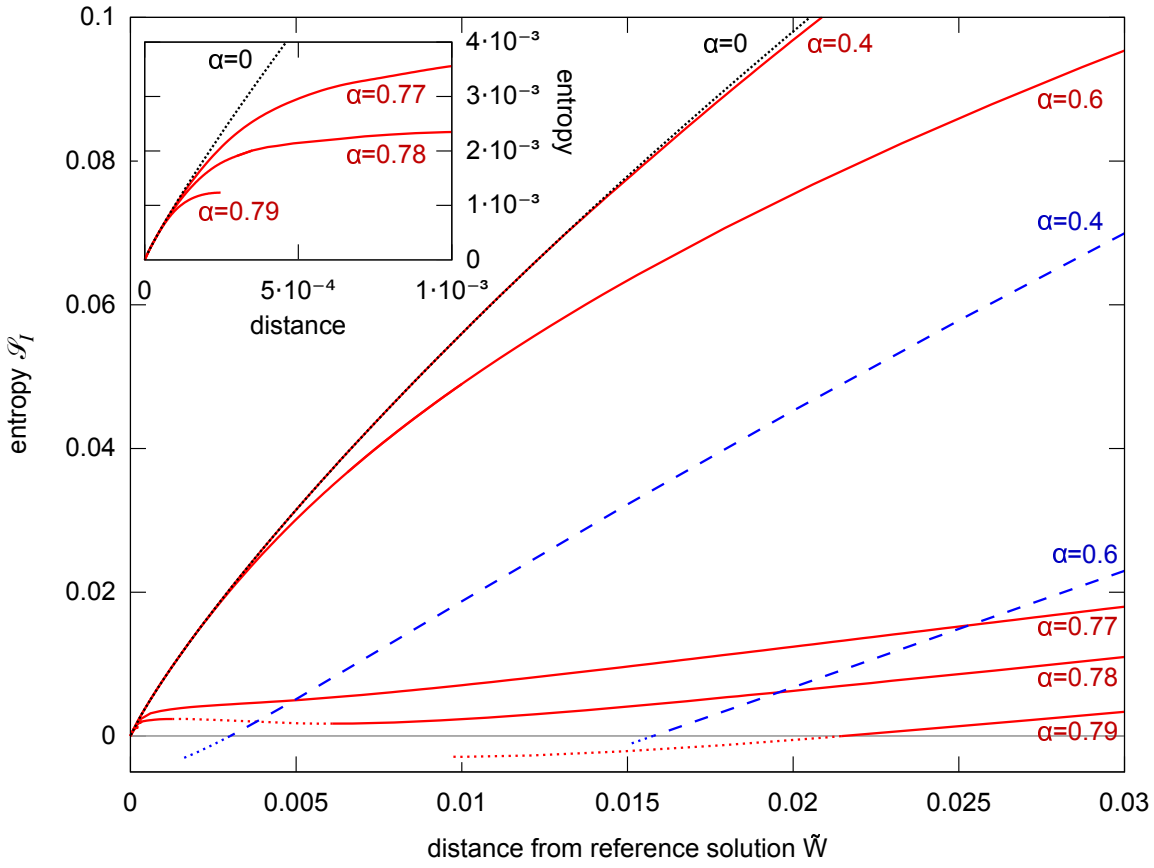

FIGURE 4.2.3. (Color online) **Large deviation analysis**. Local entropy curves at varying distance d from the reference solution $\tilde{W}$ for various α (classification case). Black dotted curve: $\alpha = 0$ case (upper bound). Red solid curves: RS results from Eq. (4.2.2) (optimal $\tilde{W}$). Up to $\alpha = 0.77$, the curves are monotonic. At $\alpha = 0.78$, a region incorrectly described within the RS Ansatz appears (dotted; geometric bounds are violated at the boundaries of the part of the curve with negative derivative). At $\alpha = 0.79$, the solution is discontinuous (a gap appears in the curve), and parts of the curve have negative entropy (dotted). Blue dashed curves: equilibrium analysis (typical $\tilde{W}$) [**49**] (dotted parts are unphysical): the curves are never positive in a neighborhood of $d = 0$. *Inset*: zoom of the region around $d = 0$ (notice the solution for $\alpha = 0.79$, followed by a gap).

(3) Up to a certain α_U (where $\alpha_U \simeq 0.77$ in the classification case and $\alpha_U \simeq 1.1$ in the generalization case), the $\mathscr{S}_I(d)$ curves are monotonic in d. Beyond α_U, there is a transition in which there appear regions of d (dotted in Fig. 4.2.3) which are not correctly described by the RS Ansatz (as for the violation of geometric bounds). It is reasonable to think that this region should be described with a higher level of Replica Symmetry Breaking (RSB). This transition

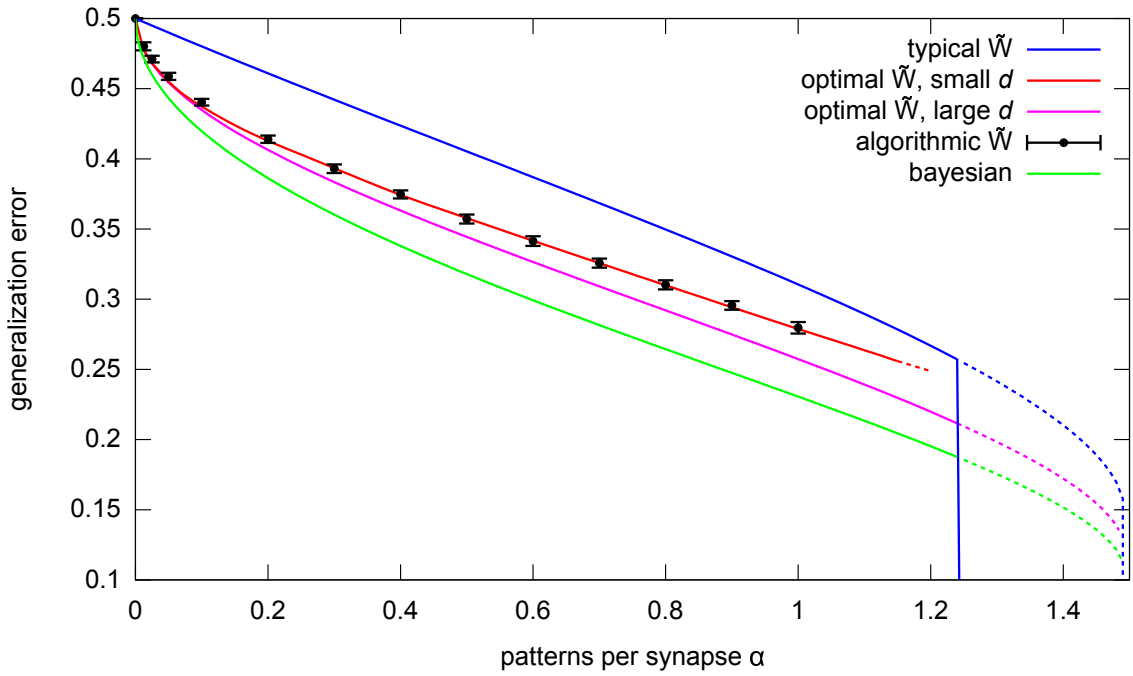

FIGURE 4.2.4. (Color online) **Generalization error (teacher-student scenario)**. From top to bottom: (blue) typical solution; (red) optimal $\tilde{W}$ from eq. (4.2.2) at $d = 0.025$ (this solution disappears after $\alpha \simeq 1.2$); (black points) solutions from SBPI at $N = 10001$, 100 samples per point; (magenta) optimal $\tilde{W}$ from eq. (4.2.2) at the value of d for which $\mathscr{S}_I$ is maximum (i.e. it equals the equilibrium entropy); (green) bayesian case: error from the average over all solutions. At $\alpha_{TS} = 1.245$ is the first-order transition to perfect learning; between α_{TS} and $\alpha = 1.5$ there is a meta-stable regime; the dashed parts of the curves correspond to unphysical solutions of the RS equations with negative entropy.

likely signals a change in the structure of the space of solutions: for $\alpha < \alpha_U$, the densest cores of solutions are immersed in huge connected structure; for $\alpha > \alpha_U$, this structure fractures and the dense cores become isolated and hard to find.

In the teacher-student scenario, the generalization properties of the optimal reference solutions $\tilde{W}$ are generally much better than those of typical solutions. This is clearly shown in Fig. 4.2.4, where I also show that the curve for small d is in striking agreement with that produced using solutions obtained from the SBPI algorithm. The generalization error decreases monotonically when increasing d, and it saturates to a plateau when $\mathscr{S}_I(d)$ becomes equal to the entropy of the typical solutions (point 2 above).

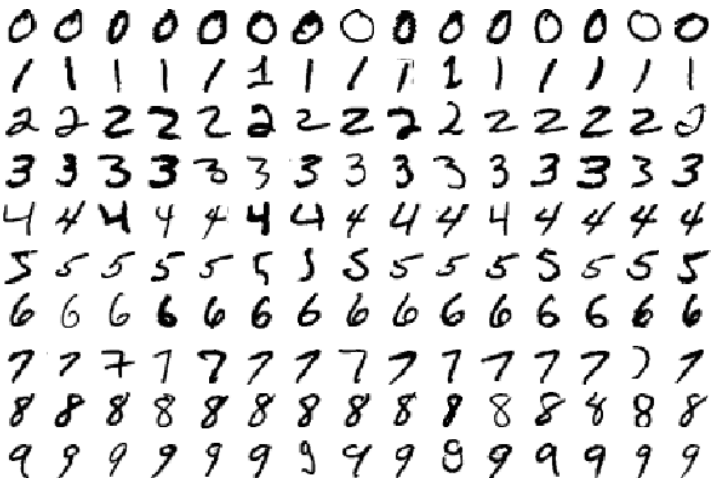

FIGURE 4.3.1. A representative subset of the MNIST training set. For each digit 15 examples are shown. The size of each image is 28x28 pixels. Each image is centered and grey level intensity is normalized.

4.3. Multi-layer network

It would be very interesting to investigate the issues presented here when confronted with more complex architectures, where single perceptrons are used to build multi-layer and multi-category classifiers. We tested a heuristic extension of the CP+R algorithm to a multi-layer classifier with L possible output labels, training these type of networks on the MNIST database benchmark [**5**], which consists of $7 \cdot 10^4$ gray-scale images of hand-written digits ($L = 10$). A subset of the training set if show in Fig. 4.3.1. In the standard benchmarking procedure, 10^4 of the images are reserved for assessing the generalization performance, and another set of 10^4 is taken as a validation set (which we do not use). The images were subject to standard unsupervised pre-processing by a Restricted Boltzmann Machine ($N = 501$ output nodes) [**3, 4**], but this is not essential for training: the inputs could be used directly, or be simply pre-processed by random projections, with only minor effects on the performance. Let us consider a multi-layer classifiers with L possible output labels. The architecture we used (Fig. 4.3.2A) consists of an array of K_2 committee machines, each comprising K_1 hidden units, whose outputs are sent to L summation nodes, and from these to a readout node which performs an argmax operation. This network therefore realizes the following map:

$$\phi\left(\xi\right) = \mathrm{argmax}_{l\in\{1,\dots,L\}}\left(\sum_{k_2=1}^{K_2} Y_{k_2 l}\,\mathrm{sign}\left(\sum_{k_1=1}^{K_1}\tau\left(W^{k_1 k_2},\xi_i\right)\right)\right)$$

where $Y_{k_2 l} \in \{-1,1\}$ are random binary weights, and $W^{k_1 k_2} \in \{-1,1\}^N$ are the synaptic weights.

The single-layer CP+R rule consists of two independent processes, a supervised one and a generalized, unsupervised one (see [**54**] for details). For the multi-layer case, we kept the unsupervised process unaltered, and used a simple scheme to back-propagate the error signals to the individual perceptron units, as follows: upon presentation of a pattern ξ whose required output is σ, in case of error ($\phi\left(\xi\right) \neq \sigma$), a signal is sent back to all committee machines which contributed to the error, i.e. all those for which $\mathrm{sign}\left(\sum_{k_1=1}^{K_1}\tau\left(W^{k_1 k_2},\xi_i\right)\right) \neq Y_{k_2\sigma}$. Each of these in turn propagates back a signal to the

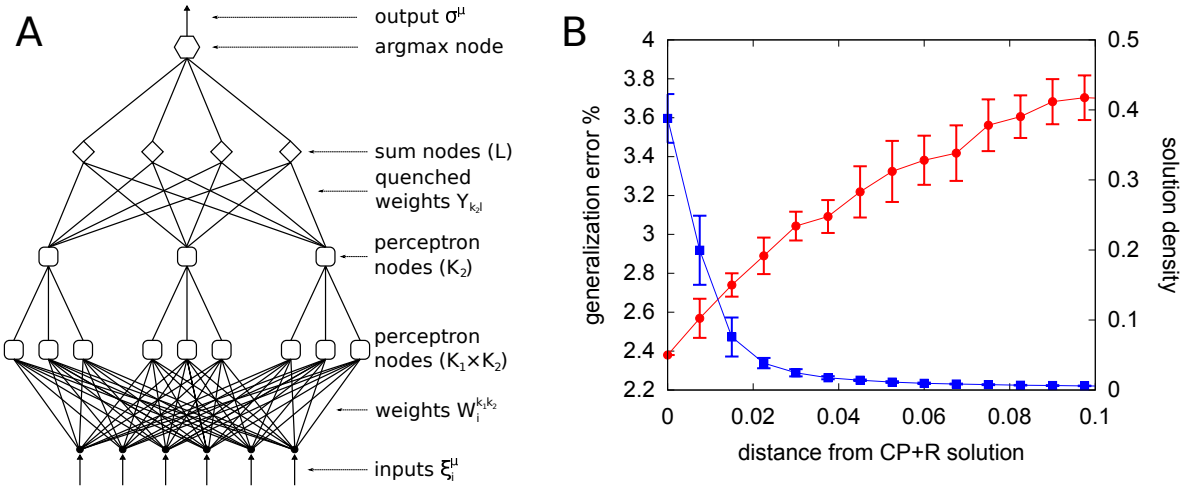

FIGURE 4.3.2. (Color online) **Multi-layer tests on MNIST**. *A*. Network scheme. *B*. results of a random walk over solutions to the training set (with $K_1 = 11$, $K_2 = 30$, $r = 0$), starting from a solution found by CP+R. Moving away from this solution, the generalization error (red, circles) increases, and the solution density (blue, squares) decreases. The same qualitative behavior is observed with all network sizes, and regardless of preprocessing.

hidden unit, among those which provided the wrong output (i.e. for which $Y_{k_2\sigma} \sum_{i=1}^{N} W_i^{k_1 k_2} \xi_i < 0$), which is the easiest to fix, i.e. for which $-Y_{k_2\sigma} \sum_{i=1}^{N} W_i^{k_1 k_2} \xi_i$ is minimum. Finally, the hidden units receiving the error signal update their internal state according to the CP+R supervised rule. We also added a "robustness" setting, such that an error signal is emitted also when $\phi(\xi) = \sigma$, but the difference between the maximum and the second maximum in the penultimate layer is smaller then some threshold r. We observed that it is indeed very easy to learn the training set with zero error, and that very good generalization errors can be reached despite the binary nature of the synapses and the fact that we did not specialize the architecture for this particular dataset. Moreover, we didn't observe any overfitting: the generalization error does not degrade by reaching zero training error, or by using larger networks. The smallest network which is able to perfectly learn the whole training dataset had $K_1 = 11$ and $K_2 = 30$, with $r = 0$; its generalization error was about 2.4%. Larger networks achieve better generalization error rates, e.g. 1.25% with $K_1 = 81$, $K_2 = 200$, $r = 120$.

As for the perceptron, we performed a random-walk process in the space of solutions, with similar results: the simplified algorithm reaches a solution which is part of a dense, large connected cluster, and the generalization properties of the starting solution are better than those of solutions found in later stages of the random walk (Fig. 4.3.2B).

4.4. Summary

In this chapter I discussed the main theoretical understanding as well as the available Belief Propagation machinery for effectively finding a solution in the perceptron learning problem. The present analysis showed that the standard equilibrium analysis does not capture the features of the types of solutions which are found by means of the BP-inspired learning protocols. The introduction

of a novel large deviation analysis revealed the existence of a dense subdominant cluster of solutions, which is the main responsible for the effectiveness of known learning algorithms, and controls their algorithmic capacities. Our results have been presented in Ref. [**57**]. More recently, some of us found that this analysis can be generalized to the case of multi-level synapses [**58**].

As I will discuss in the next chapter, these result motivated the introduction of a general Monte Carlo optimization strategy, which is also effective in the random K-SAT problem.

CHAPTER 5

Entropy Driven Monte Carlo

In chapter 4 the existence of subdominant yet accessible states in the Perceptron Learning Problem has been documented by means of analytical and computational methods. This findings inspired the introduction of a novel Markov Chain Monte Carlo (MCMC) method which explicitly seeks for states with high local entropy, and is constructed in a way that it leads the system inside a large dense cluster of zero energy configurations. The starting point of the discussion is the observation that the results obtained in last chapter are extremely robust even if one considers a slightly modified ensemble where the reference configurations are not constrained to be solutions: this motivates the introduction of a method which does not use the energy information and is guided only by the local entropy information.

It is then appealing to treat the local entropy as a pure objective function, and define a novel MCMC, which we call Entropy-driven Monte Carlo (EdMC). When applied to the binary perceptron learning problem, EdMC yields algorithmic results that are in very good agreement with the theoretical computations, and by far outperform standard Simulated Annealing in a direct comparison.

EdMC has been tested in the general context of Random Constraint Satisfaction Problems (CSPs), which offers an ideal framework for investigating the impact of the geometry of the space of solutions on the performance of sampling algorithms, and solvers as well. In many random CSPs, the computational hardness is associated to the existence of optimal and metastable states that are grouped into different clusters of nearby solutions with different sizes and properties. Large deviation analysis allows to describe in some detail the structure of such clusters, ranging from the dominant clusters (those in which one would fall by choosing uniformly at random a solution) to the subdominant ones. One possible way to search for a solution on a single instance of a random CSP is to use a Monte Carlo approach: MCMC algorithms for combinatorial optimization problems are designed to converge to a stationary distribution π that is a monotone decreasing function of the objective function one needs to minimize. A fictitious temperature is usually introduced and a Simulated Annealing (SA) procedure is employed, where this temperature is slowly decreased, to make the distribution more and more focused on the optima. Depending on the form of the stationary distribution (i.e. on the temperature) the sampling process can converge fast or it can get trapped in local minima. There is typically a tradeoff between optimality of the sampled solutions and the form of π: smooth and close to uniform distributions are the easiest to sample but the sampled configuration are most often far from optimal. On the contrary hard to sample distributions are characterized by a glassy landscape where the number of metastable minima that can trap the MCMC and break ergodicity are typically exponentially numerous.

In contrast to standard Simulated Annealing procedures, the entropy based scheme does not suffer from trapping in local minima, its smoother objective function enabling it to focus on a solution which is at the center of a dense regions of solutions. Moreover, EdMC offers a method to validate the RS calculation, and to understand better the properties of the heuristic BP-inspired algorithms that are able to find a solution to hard optimization problems.

This chapter is organized as follows: in Sec. 5.1 I introduce the unconstrained version of the RS calculation of the previous chapter, which serves as a presentation of the novel EdMC algorithm; in Sec. 5.2 EdMC is tested in two prototypical CSPs, the perceptron and random K-SAT, and its performance is systematically compared to standard Simulated Annealing.

5.1. Subdominant clusters and Entropy-driven Monte Carlo

The main idea behind Entropy-driven Monte Carlo is that the local entropy could provide a smoother landscape than the energy in a general Contraint Satisfaction Problem suffering from a proliferation of local minima, and its dynamics could lead to solutions which have the property of being surrounded by many other zero energy configurations. Computing the local entropy may be more difficult than computing the energy, but the resulting landscape may be radically different. Let us then introduce the general concept of a Constraint Satisfaction Problem and set the notation for EdMC.

5.1.0.1. *Out of equilibrium analysis of CSPs.* A generic Constraint Satisfaction Problem (CSP) can be defined in terms of N variables $x_i \in X_i$, and M constraints $\psi_\mu : D_\mu \to \{0,1\}$. Each constraint μ involves a subset $\partial\mu$ of the variables, which will be collectively represented as $x_{\partial\mu} = \{x_i : i \in \partial\mu\} \in D_\mu$, and a compatibility function ψ_μ, where $\psi_\mu(x_{\partial\mu}) = 1$ if the constraint is satisfied, 0 otherwise. For concreteness, let us focus on the case of binary spin variables $X_i = X = \{-1, +1\}$. The generalization to multi-valued variables is straightforward, and will be presented in an upcoming work for the perceptron case together with an analytical characterization [**58**]. It is costumary to define an energy function of the system simply as the number of violated constraints, namely:

$$H_0(x) = \sum_\mu E_\mu(x_{\partial\mu}) = \sum_\mu (1 - \psi_\mu(x_{\partial\mu})) \tag{5.1.1}$$

In analogy to what has been discussed in Chapter 4, I will introduce a method that counts the number of solution vectors x around a given planted vector $\tilde{x}$. To this end, *local free entropy* is defined as follows:

$$F(\tilde{x}, \gamma) = \frac{1}{N} \log \mathcal{N}(\tilde{x}, \gamma) \tag{5.1.2}$$

where

$$\mathcal{N}(\tilde{x}, \gamma) = \sum_x \prod_\mu \psi_\mu(x_{\partial\mu})\, e^{\gamma x \cdot \tilde{x}} \tag{5.1.3}$$

counts all solutions x of the CSP with a weight that depends on the distance from $\tilde{x}$. The parameter γ controls the intensity of the coupling. Solving a CSP means to find a solution vector $\tilde{x}^\star$ with zero energy, i.e. $H_0(\tilde{x}^\star) = 0$. The alternative method which will be investigated here is to optimize the function $F(\tilde{x}, \gamma)$, trying to guide the system in a region with a high density of solutions. This amounts in introducing a coupling term $\gamma\tilde{x}_i$, with $\tilde{x}_i \in \{-1, +1\}$ acting on in each variable x_i, with $\gamma \in \mathbb{R}$ the coupling strength, so that one has to consider the slightly different system described by the following Hamiltonian:

$$H(x; \tilde{x}) = H_0(x) - \gamma \sum x_i \tilde{x}_i \tag{5.1.4}$$

The directions of the external fields $\tilde{x}_i$'s are considered as external control variables, and γ sets the intensity of the external fields: in the large N limit it effectively fixes the Hamming distance d of the solutions x from $\tilde{x}$. The local free entropy $F(\tilde{x}, \gamma)$ is then obtained as the zero-temperature limit

of the free energy of the system described by $H(x;\tilde{x})$, and can be computed by Belief Propagation (see Sec. 5.1.0.2 and Appendix Sec. A.1).

The free energy of the new systems is then:

$$\mathscr{F}(\gamma, y) = -\frac{1}{Ny}\log\left(\sum_{\tilde{x}} e^{yF(\tilde{x},\gamma)}\right) \tag{5.1.5}$$

where y has the role of an inverse temperature and $-F(\tilde{x},\gamma)$ is a formal energy. In the limit of large y, this system is dominated by the ground states $\tilde{x}^\star$ with maximum local entropy; if the number of such ground states is not exponentially large in N, the *local entropy* can then be recovered by computing the Legendre transform

$$\mathscr{S}(\gamma,\infty) = -\mathscr{F}(\gamma,\infty) - \gamma S \tag{5.1.6}$$

where S is the typical value of the overlap $\frac{x\cdot\tilde{x}^\star}{N}$. The number of solutions at distance $d = \frac{1-S}{2}$ from the ground states $\tilde{x}^\star$ will then be given by:

$$e^{N\mathscr{S}(\gamma,\infty)} = \sum_x \prod_\mu \psi_\mu(x_{\partial\mu})\,\delta(NS - x\cdot\tilde{x}^\star) \tag{5.1.7}$$

For large γ (i.e. small distance d), one expects that $\tilde{x}^\star$ will eventually be in the middle of a dense cluster of solutions (provided such cluster exists) and that it is likely going to be a solution itself, a case that is strictly guaranteed only for $\gamma\to\infty$. The results discussed in the case of the perceptron highlighted the relevance of the extremal points $F(\tilde{x}^\star,\gamma)$ of the local free entropy, and showed that solutions which happen to be inside a dense *subdominant cluster* have different statistical properties with respect to the set of thermodynamically locally stable states.

5.1.0.2. *EdMC: Local entropy optimization.* Entropy-driven Monte Carlo is constructed as a simple Metropolis procedure where Monte Carlo moves are controlled by the value of the local entropy. The reason to follow this strategy instead of directly minimizing the energy $H_0(\tilde{x})$ of Eq. (5.1.1), is that it turns out that the landscape of the two objective functions is radically different: while the energy landscape can be dominated by local minima that trap local search algorithms, the local entropy landscape is much smoother. Furthermore, the behavior of this algorithm can — at least in principle — be described in the typical case with the tools of Statistical Mechanics, to the contrary of what is currently possible for the other efficient solvers, which are all based on some kind of heuristics (e.g. decimation, soft decimation a.k.a. reinforcement, etc.).

All in all, EdMC procedure is the following: $\tilde{x}$ is initialized at random; at each step $F(\tilde{x},\gamma)$ is computed by the BP algorithm; random local updates (spin flips) of $\tilde{x}$ are accepted or rejected using a standard Metropolis rule at fixed temperature y^{-1}. In practice, we found that in many regimes it suffices to use the simple greedy strategy of a zero temperature Monte Carlo ($y=\infty$). From a practical point of view, it seems more important instead to start from a relatively low γ and increase it gradually, as one would do in a classical annealing procedure. This 'scoping' procedure progressively narrows the focus of the local entropy computation to smaller and smaller regions, and will be clear (at least in the case of the perceptron) in view of analytical results in the next section.

Bethe approximation is a natural way of computing the quantity $F(\tilde{x},\gamma)$. Since it is easier to understand EdMC in terms of external fields $\tilde{x}_i$, Appendix A.1 shows a different form of the Belief Propagation equations where messages are parametrized by means of local fields, and provides the corresponding expression for the free energy.

5.2. EdMC results

EdMC has been tested in two cases, the first being the perceptron problem that originally inspired its introduction. The second is an example of a diluted problem, the Random K-SAT, which has a long tradition in the Statistical Mechanics of Optimization.

5.2.1. Perceptron. The first step was to repeat the RS computation of chapter 4 for the free energy of eq. (5.1.5), where no constraint is imposed on the reference configurations $\tilde{x}$. An important check of the plausibility of the RS results is the positivity of external entropy: in the unconstrained case, one finds that as α increases the results become patently unphysical. For example, the RS solution at $y = y^\star$ yelds a positive local entropy even beyond the critical value α_c. Therefore, the RS assumption needs to be abandoned and at least a 1-step of Replica Symmetry Breaking (1-RSB) must be studied.

There are a number of non equivalent way of breaking the Replica Symmetry, which could involve the x or $\tilde{x}$ variables. It seemed reasonable to assume that RSB occurred at the level of the $\tilde{x}$ variables, since it is expected that clusters are dense with no internal fragmentation. The details of the calculation for $y \to \infty$ are shown in Appendix A.2. The result is a system of 8 coupled equations, with α and S as control parameters (using S as control parameter instead of its conjugate γ which was used in Eq. 5.1.5 is advantageous for the theoretical analysis at high α, see below).

The external entropy which is obtained from the solutions of the equations is still negative for all values of α and S, though some important caveats have to be considered: its magnitude is much smaller than the one obtained with an RS Ansatz at $y \to \infty$; furthermore, its value goes to zero when $S \to 1$. All the other unphysical results of the RS solution were fixed at this step of RSB, and the qualitative behavior of the solution is the same as for the constrained RS version of section 4.2. As one can see in Fig. 5.2.1, the strong density at the core of the cluster is apparent also in the unconstrained case: for all α below the critical value $\alpha_c = 0.83$, the local entropy in the region of $S \to 1$ tends to the $\alpha = 0$ curve, implying that for small enough distances the region around the ground states $\tilde{x}^\star$ is extremely dense (almost all points are solutions). Analogously, there is a transition at $\alpha_U \simeq 0.77$ after which the local entropy curves are no longer monotonic, the appearance of a gap in S with no solution most likely signaling a regime with a large number of disconnected regions which originate from the fragmentation of the original cluster.

Two additional results are worth noting about the properties of the reference configurations $\tilde{x}$:

(1) In the limit $y \to \infty$, the local entropy takes exactly the same value as for the constrained case in which the $\tilde{x}$ are required to be solutions, and the same is true for the parameters that are common to both cases. The external entropy, however, is different. This is true both in the RS and the 1-RSB scenario.
(2) For the unconstrained case, the probability that the reference configuration $\tilde{x}$ makes an error on any one of the patterns is easily computed (see Fig. 5.2.2 and Appendix A.2.3). This probability is a decreasing function of S (going exponentially to 0 as $S \to 1$) and an increasing function of α. For low values of α, this probability is extremely low, such that at finite values of N the probability that $\tilde{x}$ is a solution to the full pattern set is almost 1.

Fig. 5.2.1B shows that up to $\alpha \lesssim 0.75$ we can use γ as a control parameter to determine S, while Fig. 5.2.2 shows that a solution to the learning problem could be found by maximizing the local free entropy $F(\tilde{x}, \gamma)$ (eq. (5.1.2)) as a function of $\tilde{x}$ at sufficiently large γ. This observation directly justifies the use of a scoping procedure in EdMC.

EdMC was used to test the theoretical results by searching the optimum value of the free local entropy $F(\tilde{x}, \gamma)$ at each γ for different α. As an example, Fig. 5.2.3 shows a comparison between the

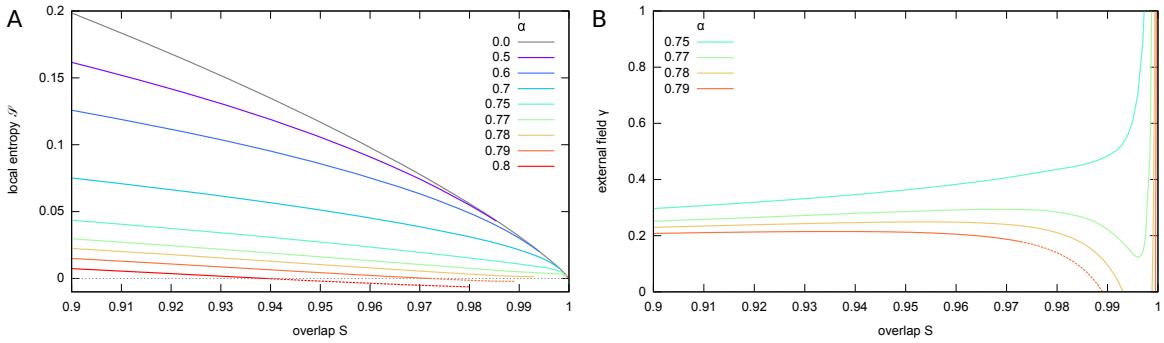

FIGURE 5.2.1. **A.** Local entropy vs overlap S, at various values of α. All curves tend to the $\alpha = 0$ case for sufficiently high S. For $\alpha \gtrsim 0.77$, a gap appears, i.e. a region of S where no solution to the saddle point equations exists. For $\alpha \gtrsim 0.79$, some parts of the curve have negative entropy (dashed). **B.** Relationship between the overlap S and its conjugate parameter, the external field γ. Up to $\alpha \lesssim 0.75$, the relationship is monotonic and the convexity does not change for all values of S; up to $\alpha \lesssim 0.77$, a solution exists for all S but the relationship is no longer monotonic, implying that there are regions of S that can not be reached by using γ as an external control parameter. The gap in the solutions that appears after $\alpha \gtrsim 0.77$ is clearly signaled by the fact that γ reaches 0; below $\alpha_c = 0.83$, a second branch of the solution always reappears at sufficiently high S, starting from $\gamma = 0$.

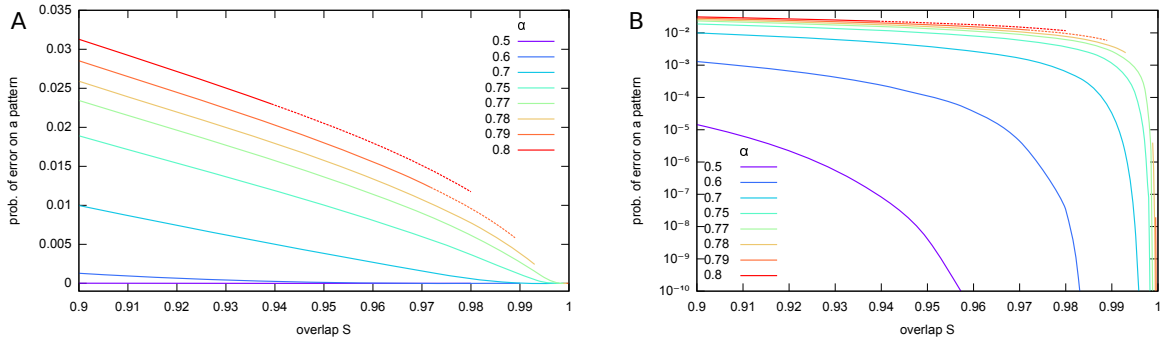

FIGURE 5.2.2. **A.** Probability of a classification error by the optimal reference configuration $\tilde{x}$, for various values of α, as a function of S. The dashed parts of the curves correspond to the parts with negative local entropy (cf. Fig. 5.2.1); the curves have a gap above $\alpha \gtrsim 0.77$. **B.** Same as panel A, but in logarithmic scale on the y axis, which shows that all curves tend to zero errors for $S \to 1$.

theoretical result and the values found by EdMC with the following procedure: EdMC was run on 2000 samples at $N = 201$ and $\alpha = 0.6$, at various values of γ (we used $\gamma = \tanh^{-1}(p)$, varying $p \in [0.4, 0.9]$ in steps of 0.1), without stopping the algorithm when a solution was found. In addition to a greedy procedure ($y = \infty$), a cooling procedure was used, in which y was initialized at 5 and increased by

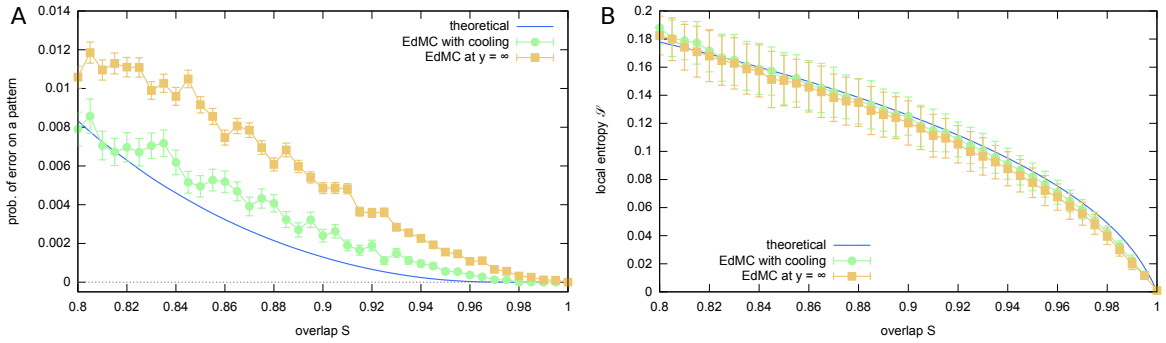

FIGURE 5.2.3. *EdMC vs theoretical results.* **A.** Probability of error on a pattern (cf. Fig. 5.2.2) **B.** Local entropy (cf. Fig. 5.2.1A). For the version with cooling, 700 pattern sets were tested for each value of γ. For the $y = \infty$ version, 2000 samples were used. Error bars represent standard deviation estimates of the mean values.

a factor of 1.01 for every 10 accepted moves. The search was stopped after $5N$ consecutive rejected moves. For each sample and each polarization level, the value of the overlap S, of the local entropy $\mathscr{S}$ (see eq. 5.1.6 above and eqs. (A.1.8) and (A.1.9) in the Appendix Sec. (A.1.1)) and of the error probability per pattern were stored. Results were then binned over the values of S, using bins of width 0.005, and local entropy and error rate are averaged in each bin. EdMC results are in good agreement with the theoretical curve: the qualitative behavior is the same (in particular: the error rate goes to zero at $S \to 1$ and the entropy is positive until $S = 1$, confirming the existence of dense clusters), the cooling procedure version yielding results which are closer to the theoretical values. For all these reasons, the 1-RSB solution seems to be a reasonable approximation to the correct solution at $y \to \infty$.

The remaining discrepancy could be ascribed to several factors: 1) finite size effects, since N is rather small; 2) inaccuracy of the Monte Carlo sampling, which would be fixed by lowering the cooling rate; 3) inaccuracy of the theoretical curve due to further steps of Replica Symmetry Breaking. Note that, with these settings, the average number of errors per pattern set is almost always less than 2 for all points plotted in Fig. 5.2.3A. Also note that, for all values of S, the mode and the median of the error distribution is at 0, and that the average is computed from the tails of the distribution (which explains the noise in the graphs). Finally note that, for all samples, points corresponding to 0 errors were found during the Monte Carlo procedure.

5.2.1.1. *EdMC vs Simulated Annealing.* The most remarkable feature of EdMC is its ability to retrieve a solution at zero temperature in a relative small number of steps, as opposed to standard Simulated Annealing. Performing a systematic comparison between the two methods for different values of α and different sized N, one finds that zero temperature MCMC immediately gets trapped in local minima at zero temperature, even at small N. In order to find a solution with MCMC a simulated annealing (SA) approach was used: inverse temperature was initially set to $y_0 = 1$, and increased by a factor f_y for every 10^3 accepted moves. The cooling factor parameter f_y was optimized on each instance. Fig. 5.2.4 shows a comparison between a typical trajectory of SA versus EdMC on the very same instance (EdMC is run at $y = \infty$ with $\gamma = \tanh^{-1}(0.6)$): at first glance, it exemplifies the typical difference in the average number of steps required to reach a solution between SA and EdMC, which is of 4 or 5 orders of magnitude for small N. Also note the smoothness of the EdMC trajectory in contrast to SA: the local entropy landscape is far smoother and ensures a rapid convergence to the

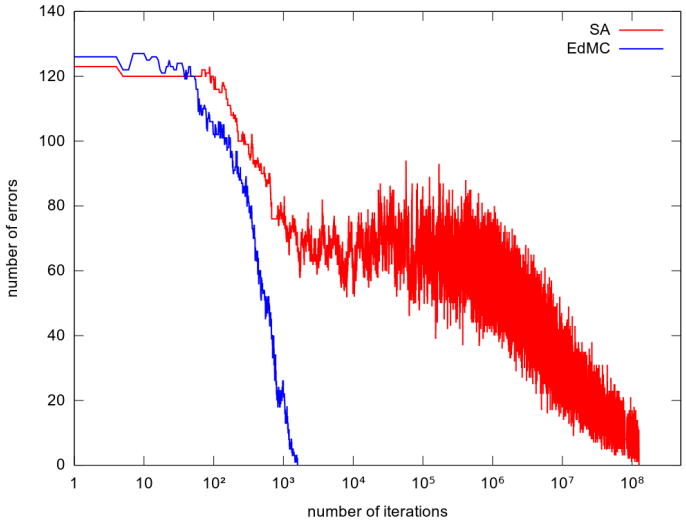

FIGURE 5.2.4. *Perceptron Learning Problem*, $N = 801$, $\alpha = 0.3$. Typical trajectories of standard Simulated Annealing (red curve, right) and Entropy-driven Monte Carlo (blue curve, left). Notice the logarithmic scale in the x axis. EdMC is run at 0 temperature with fixed $\gamma = \tanh^{-1}(0.6)$, SA is started at $y_0 = 1$ and run with a cooling factor of $f_y = 1.001$ for each 10^3 accepted moves, to ensure convergence to a solution.

region with highest density of solutions. The scaling of the two algorithms was systematically studied at $\alpha = 0.3$ and $\alpha = 0.6$, varying N between 201 and 1601 and measuring the number of iterations needed to reach a solution to the learning problem.

For the SA tests, the following procedure was used: for each instance of the problem, one tries to find a solution at some value of the cooling factor f_y; after $10^5 N$ consecutive rejected moves, a reduced f_y is introduced, and the procedure is repeated until a solution is eventually found. The values of f_y were $\{1.1,\ 1.05,\ 1.02,\ 1.01,\ 1.005,\ 1.002,\ 1.001,\ 1.0005,\ 1.0001\}$. This allowed to measure the least number of iterations required by SA to solve the problem (the shown scaling plots only report the number of iterations for the last attempted value of f_y). At $\alpha = 0.3$, all tested instances were solved, up to $N = 1601$. At $\alpha = 0.6$, however, this procedure failed to achieve 100% success rate even for $N = 201$ in reasonable times; at $N = 401$, the success rate was 0% even with $f_y = 1.0001$; at $N = 1601$, using $f_y = 1.0001$ did not seem to yield better results than $f_y = 1.1$. Therefore, no data is available for SA at $\alpha = 0.6$.

For the EdMC tests, the following procedure was used: the algorithm was started at $\gamma = \tanh^{-1}(p)$ with $p = 0.4$, and the Monte Carlo procedure was run directly at $y = \infty$; if a solution was not found, p was increased by a step of 0.1 and continued from the last accepted configuration, up to $p = 0.9$ if needed. This procedure converges to a solution in all cases.

The results are shown in Fig. 5.2.5, in log-log scale. For SA (panel A in the figure), the behavior is clearly exponential at $\alpha = 0.3$, and missing altogether for $\alpha = 0.6$. For EdMC (panel B in the figure), the data is well fit by polynomial curves, giving a scaling $\sim N^{1.23}$ for $\alpha = 0.3$ and $\sim N^{1.74}$ for $\alpha = 0.6$. Also note the difference of several orders of magnitude in the ranges of the y axes in the two panels.

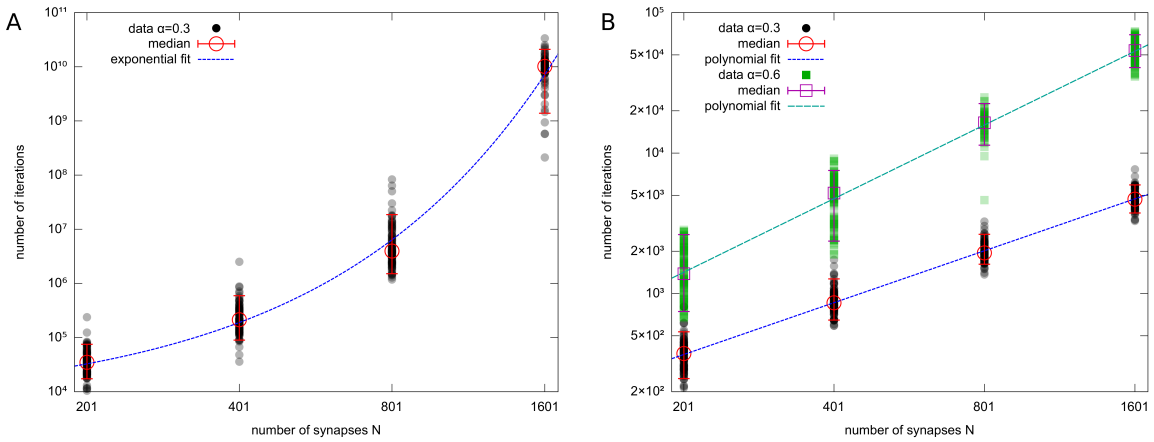

FIGURE 5.2.5. *Perceptron Learning Problem.* Number of iterations required to reach 0 energy in log-log scale, as a function of the problem size N. **A:** Simulated Annealing at $\alpha = 0.3$, **B:** EdMC at $\alpha = 0.3$ (bottom) and $\alpha = 0.6$ (top). See text for the details of the procedure. Notice the difference in the y axes scales. For both methods, 100 samples were tested for each value of N. Color shades reflect data density. Empty circles and squares represent medians, error bars span the 5-th to 95-th percentile interval. The dashed lines are fitted curves: the SA points are fitted by an exponential curve $\exp(a + bN)$ with $a = 8.63 \pm 0.06$, $b = (8.79 \pm 0.08) \cdot 10^{-3}$; the EdMC points are fitted by two polynomial curves aN^b with $a = 0.54 \pm 0.04$, $b = 1.23 \pm 0.01$ for $\alpha = 0.3$, and with $a = 0.14 \pm 0.02$, $b = 1.74 \pm 0.02$ for $\alpha = 0.6$.

5.2.2. A different example: K-SAT. Random K-SAT has been central in the development of the Statistical Mechanics approach to CSPs [**50, 2**]. Here I just want to point out that, when varying the number of constraints per variable α, the problem undergoes a sequence of phase transitions, related to the fragmentation of the phase space in a huge number of disconnected clusters of solutions. This rich phenomenology, observed well below the UNSAT threshold (above which no solution typically exists at large N), can be analyzed by the cavity method in the framework of 1-step Replica Symmetry Breaking (1-RSB), and is reflected in the exponential slowing down of greedy algorithms as well as sampling strategies.

The satisfiability problem, in its 'random K-SAT' instantiation, consists in finding an assignment for N truth values that satisfies $M = \alpha N$ random logical clauses, each one involving exactly K different variables. Let us then consider N Boolean variables $\{t_i \in \{0, 1\}\}$, with the common identification $\{0 \to \text{FALSE}, 1 \to \text{TRUE}\}$. A given clause μ is the logical OR of its variables, whose indices are $i_1^\mu, ..., i_K^\mu$, and which can appear negated. Let us work in a spin representation $x_i = 2t_i - 1$ and introduce the couplings $J_{i_r}^\mu \in \{-1, +1\}$, where $J_{i_r}^\mu = 1$ if the variable $x_{i_r^\mu}$ appears negated in clause μ, and $J_{i_r}^\mu = -1$ otherwise. The graph structure is random, in that each clause involves K variables extracted uniformly at random, and the couplings are also unbiased i.i.d. random binary variables. With these definitions, the solutions to a K-SAT problem are the zero energy configurations of the

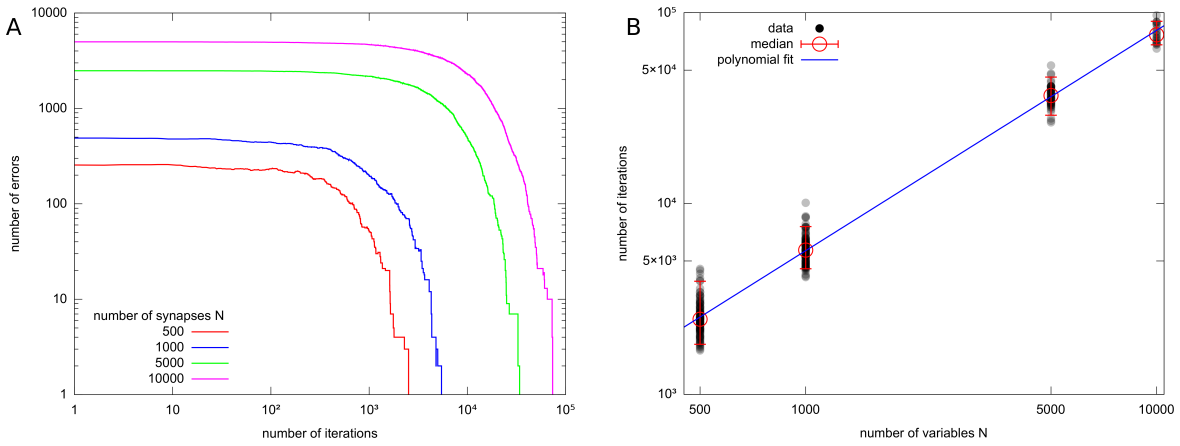

FIGURE 5.2.6. *Random 4-SAT*, $\alpha = 8.0$, EdMC results with $y = \infty$ and $\gamma = \tanh^{-1}(0.3)$. **A.** Typical trajectories at different values of N, from 500 to 10000 (bottom to top), in log-log scale. **B.** Number of iterations to reach 0 energy, in log-log scale, as a function of the problem size N. Color shades reflect data density. Empty circles represent medians, error bars span the 5-th to 95-th percentile interval. The data is well fitted by a polynomial curve aN^b (blue line), with $a = 1.94 \pm 0.15$, $b = 1.15 \pm 0.01$.

following Hamiltonian:

$$H_{\text{SAT}} = 2 \sum_{\mu=1}^{M} \prod_{k=1}^{K} \left(\frac{1 + J_{i_r}^{\mu} x_{i_r^{\mu}}}{2} \right) \tag{5.2.1}$$

which counts the number of violated clauses.

The results displayed so far for the Binary Perceptron rely on a general scheme that can be applied in principle to other CSPs or optimization problems. The main bottleneck in implementing such extensions resides in the possibility of computing the local entropy efficiently, e.g. by BP or some other sampling technique. As proof of concept, EdMC was applied to the very well studied case of random K-SAT [**59, 60, 61**] focusing on the non trivial case $K = 4$, at various N and α. Random 4-SAT is characterized by three different regimes[**62, 63**]: For $\alpha < \alpha_d = 9.38$ the phase is RS and the solution space is dominated by a connected cluster of solutions with vanishing correlations among far apart variables. For $\alpha_d < \alpha < \alpha_c = 9.47$ the dominant part of the solution space breaks into an exponential number of clusters that have an extensive internal entropy. Long range correlations do not vanish. For $\alpha_c < \alpha < \alpha_s = 9.931$ the solution space is dominated by a sub-exponential number of clusters. Eventually for $\alpha > \alpha_s$ the problem becomes unsatisfiable. The hard region for random 4-SAT is $\alpha \in [\alpha_d, \alpha_s]$, i.e. where long range correlations do not vanish. In such region SA are expected to get stuck in glassy states and most of the heuristics are known to fail.

In the RS regime, EdMC succeeds in finding a solution in a small number of steps, confirming the smooth nature of the objective function. Typical trajectories for different values of N are depicted in Figure 5.2.6A. Figure 5.2.6B shows the scaling behavior with N, which is polynomial ($\sim N^{1.15}$) as in the case of the Perceptron. In the hard phase the method suffers from the lack of convergence of BP.

Even if BP (technically the 1-RSB solution with $m = 1$) would converge up to the condensation point α_c, the addition of the external fields prevent BP from converging even below such point. Instead of resorting to a 1-RSB cavity algorithm to compute the local entropy, a simple heuristic strategy was adopted, just to show that the overall method is effective.

When BP does not converge, a good estimate for the free energy F is its average $\bar{F}$ over a sufficient number of BP iterations. While this trick typically leads to a solution of the problem, it has the drawback of making the overall Monte Carlo procedure slow. It turns out, though, that there is much more information in the BP cavity marginal which could effectively be used to choose the Monte Carlo step to guide the system into a state of high local density of solutions. Eventually the heuristic turns out to be extremely fast and capable of solving hard instances up to values of α close to the SAT/UNSAT threshold. The same heuristic has also been tested in the Perceptron learning problem with excellent results at high values of α.

The main step of the heuristic method consists in performing an extensive number of flipping steps at each iteration in the direction of maximum free energy, choosing the variables to flip from the set of x_i's whose cavity marginals *in absence* of the external field h_i are not in agreement with the direction $\tilde{x}_i$ of the field itself (a precise definition of the marginals h_i is given in Appendix A.1). Let us call V the set of such variables, in a ranked order with respect to the difference between the external and the cavity fields, such that the ones with the largest difference come first. In the spirit of MCMCs, a collective flip of all the V variables is proposed, and the new value of F is computed. The collective flip is always accepted if there is an increase in free energy, otherwise it is accepted with probability $e^{y\Delta F}$, where ΔF is the free energy difference. When a collective flip is accepted, a new set V is computed. If, on the contrary, the flips are rejected, a new collective flip is proposed, simply eliminating the last variable in the ranked set V, and the procedure is repeated until the set is empty. In the general case this procedure quickly leads to a solution with zero energy. If all the collective flips in V are rejected, the procedure is terminated, and a standard EdMC is started from the last accepted values of $\tilde{x}_i$.

As it turns out, most of these collective moves are accepted immediately. The interpretation is that these moves try to maximize, at each step, the local contributions F_i's associated to each variable x_i in the Bethe free energy, in presence of an external field $\gamma\tilde{x}_i$. As for standard EdMC, an annealing strategy in y can be used, as well as a 'scoping' strategy, i.e. a gradual increase of the external field γ as well. The resulting algorithm is detailed in Appendix A.3. Scoping appears to be more fundamental than annealing in y, and in many cases crucial to produce a quick solution: as γ is increased, smaller regions are progressively observed, and this focus can eventually lead the search into a given compact cluster of solutions, a region sufficiently dense so that Replica Symmetry locally holds. Indeed, in the typical observed trajectory of this algorithm in the presence of the scoping dynamics, even when BP suffers from convergence issues in the first steps, convergence is restored when the external fields vector $\tilde{x}$ gets close to a region of high solution density. In Fig. 5.2.7 the performance of the heuristic algorithm in the hard phase of K-SAT is exemplified by the probability of finding a solution at high α (the scoping and annealing parameters have been properly tuned so as to optimize the probability of success).

5.3. Summary

In Chapter 4 I dealt with an appropriate out-of-equilibrium measure in the theoretical analysis of subdominant clusters in the perceptron. Motivated by the previous analysis, a novel method for efficiently sampling over solutions in single instances of CSPs has been introduced.

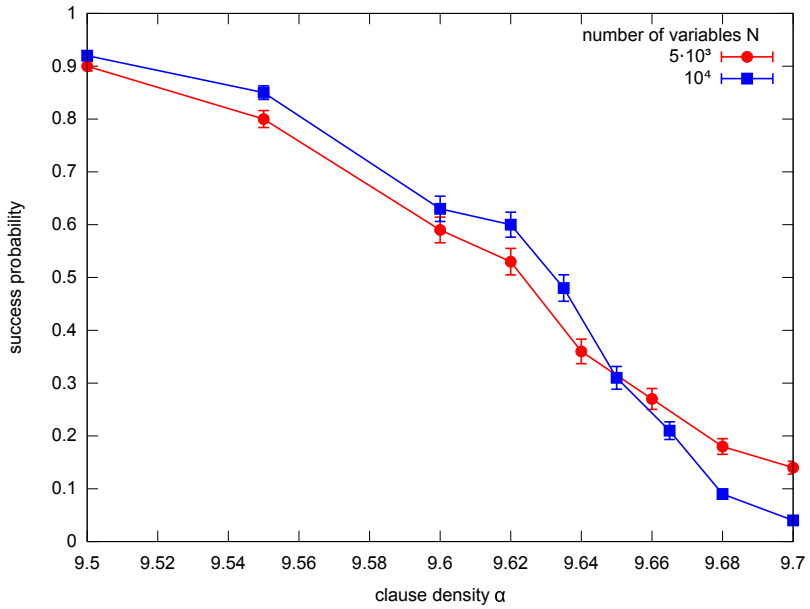

FIGURE 5.2.7. Probability of finding a solution for the faster (heuristic) EdMC algorithm on random instances of the 4-SAT problem as a function of the clause density α. Data points are obtained using Algorithm in A.3 (but without resorting to a standard EdMC as the set V of variables to be flipped is empty; rather, the algorithm is stopped and considered to have failed in this case) and averaging over 100 samples for each value of α and each problem size. For simplicity, the parameters of the algorithm are fixed once for all simulations, even though they could be fine-tuned to achieve better performance: scoping coefficient $f_\gamma = 1.05$, annealing coefficient $f_y = 1.1$ and starting values $\gamma_0 = 0.1,\ y_0 = 10^{-2}$.

At variance with standard Monte Carlo methods, minimization of energy naturally emerges from the maximization of local entropy, a quantity that can be easily estimated by means of Belief Propagation and that effectively guides an MCMC straight into a region with a high density of solutions, thus providing an efficient solver. The striking difference in performance between SA and EdMC can be understood in terms of a smoother landscape of local entropy as opposed to the energy: if we compare a zero temperature EdMC to an energy guided SA with low cooling rate, the former is orders of magnitude faster in terms of attempted moves, and does not suffer from trapping in local minima. The results described in this chapter have been recently accepted for publications [**64**].

Even if the use of Belief Propagation is necessary for the current implementation of EdMC, some of us are currently testing a new method for constructing an estimate of the local entropy that is based on a different Monte Carlo approach. Interestingly enough, this method also sheds light on the intimate nature of the standard reinforcement procedure that is able to turn Belief Propagation in a practical solver in single instances. A further discussion is beyond the scope of the present Thesis, and will be treated elsewhere.

CHAPTER 6

Inverse Dynamics in epidemics: the SIR model

Identifying the origin of an epidemic outbreak is a challenging open problem in epidemiology, which has gained a lot of attention in recent years [**65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 15, 75**]. As I will briefly point out in the following, the so called *patient zero* problem represents a particularly hard inverse problem even for simple stochastic epidemic models, such as the Susceptible-Infected (SI) model and the more general Susceptible-Infected-Recovered (SIR) model.

Epidemic models provide a simple description of the mechanisms behind disease transmission [**76**], and can be employed to investigate the sources of large epidemic outbreaks, as well as to analyze the dynamical properties of disease spreading over networks. It is now feasible to obtain complete data on true contacts between individuals in real time, and to construct reliable networks of effective contacts. Moreover, SIR model has a broad spectrum of applications: apart from epidemiology, it can be considered as a good model for the spreading of viruses over networks of computers, as well as rumors over social network, and its use can thus be profitable in all those situations where the static structure of a network is easily accessible.

In the next 3 chapters, I will deal with the Susceptible-Infected-Recovered (SIR) [**77**] model, which is considered to provide a good description for those diseases in which the person contracting the disease becomes immune to future infections after recovery, such as measles, rubella, chicken pox and generic influenza. The same model can be applied to lethal diseases, such as HIV or Ebola, provided that the recovered state is replaced by a removed state.

The *patient zero* problem on a network with a SIR dynamics can be cast as a maximum likelihood estimation problem. In general, estimating the maximum of a properly defined likelihood function corresponds to solve a (generally non-convex) optimization problem in the space of all possible epidemic propagations that are compatible with the data. A maximum likelihood estimator was proposed by Shah and Zaman [**65, 67**] for propagations with a unique source on regular trees, under the name of *rumor centrality* (see also [**71**]) and later extended to probabilistic observations in Ref. [**74**]. On general graphs, the number of propagation paths grows exponentially with the number of nodes, making the exact inference infeasible in practice. Instead of evaluating the likelihood function, Zhu and Ying put forward a method to select the path that most likely leads to the observed snapshot [**73**]. For general graphs, other heuristic inference methods are based on centrality measures [**66, 68**], on the distance between observed data and typical outcome of propagations for given initial conditions [**70**] or on the assumption that the epidemic propagation follows a breadth-first search tree [**69, 72**]. Even fewer results exist for epidemic inference with multiple sources [**72**].

Karrer and Newman [**78**] were the first to formulate a Message Passing approach to the forward problem, i.e. the characterization of the probability distribution of the infection paths on a given network. This approach was later used by the authors of Ref. [**75**], which constructed a maximum likelihood procedure for the source detection, the so called Dynamic Message Passing (DMP). The limit of their approach resides in the assumption of a likelihood function that is factorized over sites,

which is not necessarily consistent with the more accurate underlying approximation introduced in [**78**].

Our group recently proposed a Belief Propagation (BP) approach, described in detail in section 6.1, that does not make use of a mean field factorization assumption [**15**]. The main ingredient, which will be at the heart of subsequent developments described in the following chapters, is that of building a static model for the description of the stochastic dynamics. In a Bayesian framework, one is confronted with the problem of evaluating the posterior distribution over the sources of infection, a task which can be tackled with the use of Belief Propagation, provided some sparsity assumptions on the structure of the original graph are satisfied. The BP approximation, which gives the exact maximum likelihood solution on trees, turned out to work very well also on general graphs, outperforming other methods on many graph topologies, in the presence of one or more sources, and also when observations are by large extent incomplete.

My work has been focused on generalizing the inference method to take into account a setting in which observations happen to be affected by some kind of noise, i.e. there is a non-zero probability of observing each individual in a state which is different from its true one. This is a common problems in epidemiology, given the considerable amount of false negatives and positives in the gathering of epidemiological data [**79, 80**]. More generally, the present approach can easily incorporate the interesting case in which recovered and susceptible individuals are not distinguishable: this case was recently considered in Ref. [**73**], where the authors employ the most likely infection path method on trees, and similar heuristics on general graphs.

In what follows, I will firstly introduce a representation of the stochastic epidemic dynamics in terms of a graphical model. Belief propagation equations will be derived in section 3 (a useful efficient implementation is discussed in Appendix B). In section 4 the Bayesian inference method is tested on different random graphs. Section 5 is devoted to an extension of the original BP method that allows to infer the epidemic parameters while performing the source identification task. I will elaborate more on an interesting generalization of this two-step inference approach in chapter 8.

6.1. Graphical model representation of the epidemic process

Let us consider a graph $G = (V, E)$ that stands for the contact network of a set V of individuals. Each node i is attached to a time dependent state variable $x_i^t \in \{S, I, R\}$ so that at time t it can be in one of three possible states: susceptible (S), infected (I), and recovered/removed (R). I will focus on the discrete time version of the SIR model, which is a Markovian stochastic process. At each time step (e.g. a day), if node i is infected it has a finite probability λ_{ij} to spread the disease to each of his neighbors j; at the same time, i can recover with probability μ_i. Once recovered, individuals do not get sick anymore, and are unable to spread the disease further. Let us note that one recovers two interesting well known models as special cases simply putting $\mu_i \equiv 1$ (independent cascades model [**81**]) or $\mu_i \equiv 0$ (SI model). The transition probabilities of the SIR process satisfy the simple factorization relation $P(\mathbf{x}^{t+1}|\mathbf{x}^t) = \prod_i P(x_i^{t+1}|\mathbf{x}^t)$ where

$$
\begin{aligned}
P(x_i^{t+1} = S|\mathbf{x}^t) &= \mathbb{I}[x_i^t = S] \prod_{j\in\partial i} (1 - \lambda_{ji}\mathbb{I}[x_j^t = I]) \\
P(x_i^{t+1} = I|\mathbf{x}^t) &= (1-\mu_i)\mathbb{I}[x_i^t = I] + \mathbb{I}[x_i^t = S](1 - \prod_{j\in\partial i} (1 - \lambda_{ji}\mathbb{I}[x_j^t = I])) \\
P(x_i^{t+1} = R|\mathbf{x}^t) &= \mu_i\mathbb{I}[x_i^t = I] + \mathbb{I}[x_i^t = R].
\end{aligned}
$$

The inherent irreversibility of the SIR process can be exploited to find a simple parametrization of spreading paths that turns out to be suitable for the sake of the inference procedure: let us call $t_i = \min\{t : x_i^t = I\}$ the *infection time* of node i and $g_i = \min\{g : x_i^{t_i+g+1} = R\}$ his *recovery time* (individual i recovers at time t_i+g_i): a realization of the stochastic process is fully specified by knowing t_i's, g_i's for each individual, as well as all the *transmission delay* s_{ij}, i.e. the time that takes to node i to trasmit his disease to j once he got infected.

For a given initial configuration $\{x_i^0\}$, one first draws randomly the recovery time g_i of each node i and an infection *transmission delay* s_{ij} from node i to node j, for all pairs (ij). The recovery times $\{g_i\}$ are independent geometrically distributed random variables

$$\mathcal{G}_i(g_i) = \mu_i (1-\mu_i)^{g_i}, \tag{6.1.1}$$

while the delays $\{s_{ij}\}$ are extracted from the following truncated geometric distribution,

$$\omega_{ij}(s_{ij}|g_i) = \begin{cases} \lambda_{ij}(1-\lambda_{ij})^{s_{ij}}, & s_{ij} \le g_i \\ \sum_{s>g_i} \lambda_{ij}(1-\lambda_{ij})^{s}, & s_{ij} = \infty, \end{cases}$$

Note that the value $s_{ij} = \infty$ contains the mass of the distribution beyond the hard cut-off g_i imposed by the recovery time. For fixed valued of g_i and s_{ij}, the relationship between infection times is purely deterministic, and can be expressed in N hard dynamical constraints with the following form

$$t_i = 1 + \min_{j\in\partial i}\{t_j + s_{ji}\} \tag{6.1.2}$$

Let us then write the conditional distribution of *infection* and *recovery times* for a given initial condition:

$$\begin{aligned} \mathcal{P}(\mathbf{t},\mathbf{g}|\mathbf{x}^0) &= \sum_{\mathbf{s}} \mathcal{P}(\mathbf{s}|\mathbf{g})\,\mathcal{P}(\mathbf{t}|\mathbf{x}^0,\mathbf{g},\mathbf{s})\,\mathcal{P}(\mathbf{g}) \\ &= \sum_{\mathbf{s}} \prod_{i,j} \omega_{ij}(s_{ij}|g_i) \prod_i \phi_i(t_i,\{t_k,s_{ki}\}_{k\in\partial i})\mathcal{G}_i(g_i), \end{aligned} \tag{6.1.3}$$

where

$$\phi_i(t_i,\{t_k,s_{ki}\}_{k\in\partial i}) = \delta(t_i, \mathbb{I}[x_i^0 \neq I](1+\min_{k\in\partial i}\{t_k+s_{ki}\})) \tag{6.1.4}$$

is a characteristic function which imposes on each node i the dynamical constraint (6.1.2). Should one have some prior knowledge on the density of network states at initial time t_0, it can be easily implemented in terms of a prior $\mathcal{P}(\mathbf{x}^0) = \prod_i \gamma_i(x_i^0)$ factorized over sites, with

$$\gamma_i(x_i^0) = \gamma\delta(x_i^0, I) + (1-\gamma)\delta(x_i^0, S) \tag{6.1.5}$$

and γ a small constant.

Let us now suppose that one has access to the whole state of the network at a given time T, and is interested in the posterior probability of the initial states of each node. The posterior distibution reads

$$\mathcal{P}(\mathbf{x}^0|\mathbf{x}^T) \propto \sum_{\mathbf{t},\mathbf{g}} \mathcal{P}(\mathbf{x}^T|\mathbf{t},\mathbf{g})\,\mathcal{P}(\mathbf{t},\mathbf{g}|\mathbf{x}^0)\,\mathcal{P}(\mathbf{x}^0) \tag{6.1.6}$$

$$= \sum_{\mathbf{t},\mathbf{g},\mathbf{s}} \prod_{i,j} \omega_{ij} \prod_i \phi_i \mathcal{G}_i \gamma_i \zeta_i^T \zeta_i^0 \tag{6.1.7}$$

where $\mathbf{x}^t$ is a deterministic function of the set of infection and recovery times $(\mathbf{t},\mathbf{g})$, so that

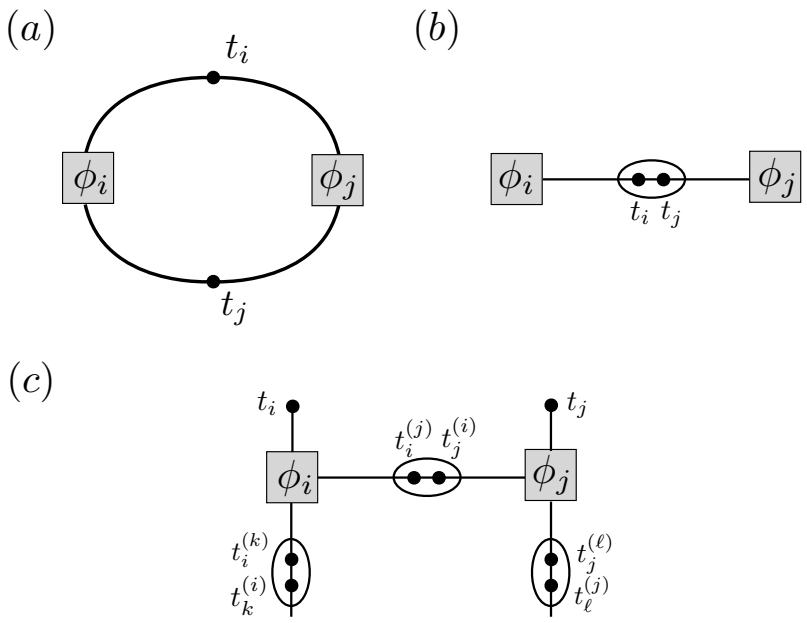

FIGURE 6.1.1. (a) Example of a loopy factor graph representations induced by constraints such as those in (6.1.2). (b) Disentangled factor graph. (c) A more convenient representation of the disentangled factor graph. For simplicity, the dependency on $\{s_{ij}\}$ is not considered.

$$\mathcal{P}\left(\mathbf{x}^T|\mathbf{t},\mathbf{g}\right) = \prod_i \zeta_i^T\left(t_i, g_i, x_i^T\right) \tag{6.1.8}$$

$$\mathcal{P}\left(\mathbf{t},\mathbf{g}|\mathbf{x}^0\right) \propto \prod_i \zeta_i^0\left(t_i, g_i, x_i^0\right) \tag{6.1.9}$$

with

$$\zeta_i^t = \mathbb{I}\left[x_i^t = S, t < t_i\right] + \mathbb{I}\left[x_i^t = I, t_i \le t < t_i + g_i\right] + \mathbb{I}\left[x_i^t = R, t_i + g_i \le t\right]. \tag{6.1.10}$$

One is now confronted with a complicated generative model, the objective being that of computing single site marginals $\mathcal{P}\left(x_i^0|\mathbf{x}^T\right)$ from the posterior distribution in 6.1.7. The problem of computing the marginals from (6.1.7) is in general intractable (NP-hard). The authors of [**15**] tackled this problem by means of Belief Propagation: it thus provided the exact results to the maximum likelihood problem on acyclic graphs, and it was shown to perform very well by means of simulations on random graphs of various types and on real contact networks. The first step is to introduce a representation of (6.1.7) in terms of a factor graph, with variable nodes represented by circles and factor nodes represented by squares. If one observes (6.1.2), it is immediately apparent that a pair of variable nodes corresponding to the infection times t_i and t_j of neighboring individuals are involved in the two factors ϕ_i and ϕ_j. If

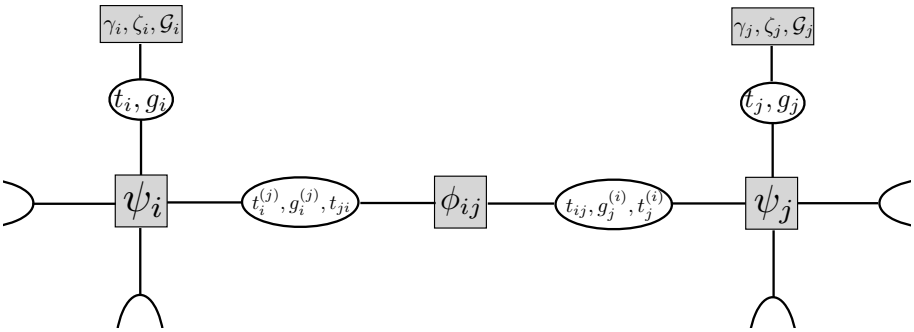

FIGURE 6.1.2. Factor graph representation of the graphical model associated with the distribution (6.1.12).

one then follows the naive recipe for the construction of the factor graph, the resulting graph of (6.1.7) has a huge number of short local loops (see Fig.6.1.1a). In a previous work [**82, 83**], it has been shown that a good way of getting rid of short loops is to first group pairs of infection times (t_i, t_j) in the same variable node as in Fig.6.1.1b, and then introduce a set of copies $t_i^{(j)}$ of the infection time t_i for each edge (i, j). The latter will be forced to take the common value t_i (see Fig.6.1.1c) by including the constraint $\prod_{k\in\partial i} \delta(t_i^{(k)}, t_i)$ in the factor ϕ_i. For convenience, all variable nodes $\{t_i\}$ are kept in the factor graph. One follows the same procedure for the g_i's variables, introducing the copies $g_i^{(j)}$ and $g_j^{(i)}$ and imposing the constraint $\prod_{k\in\partial i} \delta(g_i^{(k)}, g_i)$ for each node i.

More importantly, the factors ϕ_i depend on infection times and transmission delays just through the sums $t_i^{(j)} + s_{ij}$, so that one may naturally introduce the variables $t_{ij} = t_i^{(j)} + s_{ij}$, thus dropping the s_{ij} explicit dependence. With the modified factor graph representation in mind, the posterior distribution may be written as

$$\mathcal{P}\left(\mathbf{x}^0|\mathbf{x}^T\right) \propto \sum_{\mathbf{t},\{t_{ij}\},\mathbf{g}} \mathcal{Q}(\mathbf{g}, \mathbf{t}, \{t_{ij}\}, x_0) \tag{6.1.11}$$

with

$$\mathcal{Q}(\mathbf{g}, \mathbf{t}, \{t_{ij}\}, x_0) = \frac{1}{Z} \prod_{i<j} \phi_{ij} \prod_i \psi_i \mathcal{G}_i \gamma_i \zeta_i^T \zeta_i^0. \tag{6.1.12}$$

with the introduction of the new factors

$$\phi_{ij} = \omega_{ij}(t_{ij} - t_i^{(j)}|g_i^{(i)})\omega_{ji}(t_{ji} - t_j^{(i)}|g_j^{(i)}) \tag{6.1.13}$$

and

$$\begin{aligned} \psi_i &= \delta(t_i, \mathbb{I}[x_i^0 \neq I](1 + \min_{j\in\partial i}\{t_{ji}\})) \prod_{j\in\partial i} \delta(t_i^{(j)}, t_i)\delta(g_i^{(j)}, g_i) \\ &= \phi_i(t_i, \{t_{ji}\}_{j\in\partial i}) \prod_{j\in\partial i} \delta(t_i^{(j)}, t_i)\delta(g_i^{(j)}, g_i). \end{aligned} \tag{6.1.14}$$

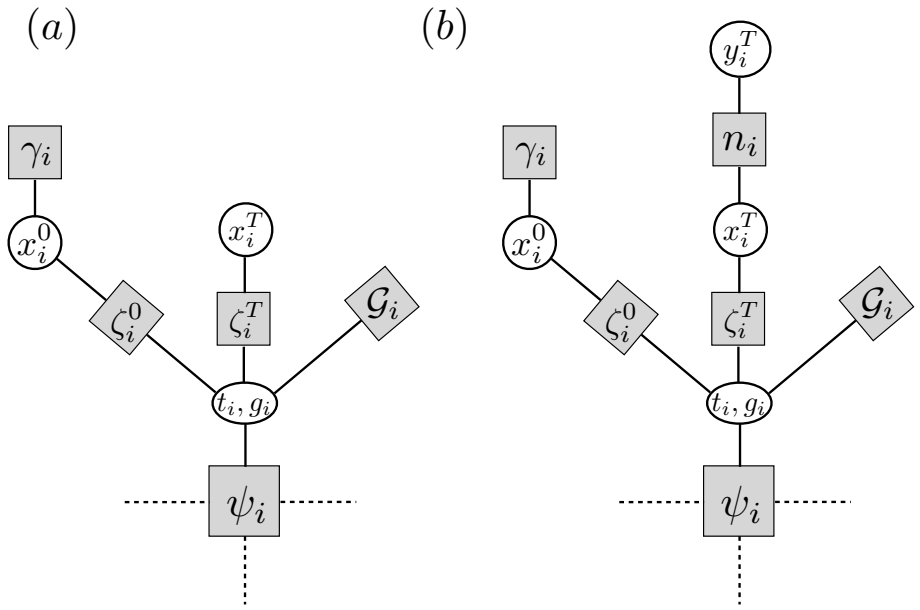

FIGURE 6.1.3. (a) Detailed description of the internal structure of the factor node containing $\gamma_i, \zeta_i^0, \zeta_i^T, \mathcal{G}_i$ in Fig.6.1.2. (b) The modified factor graph used to include the observation models discussed in Section 6.3.

The factor graph representation of (6.1.12) is shown in Figure 6.1.2. The factor node grouping $\xi_i, \zeta_i^0, \zeta_i^T$ and $\mathcal{G}_i$ has in fact a more complex structure that is described in detail in Fig.6.1.3a: the functions ζ_i^0, ζ_i^T, defined in (6.1.10), connect the infection time t_i with the states x_i^0 and x_i^T of the node i at the initial and final times, γ_i is the prior on the initial state, while $\mathcal{G}_i(g_i)$ is the recovery time distribution.

The factor graph now has the same topology of the original graph G: BP equations give the exact solution if G is a tree, and can be also profitably used on general graphs with loops. Once single posterior marginals $\mathcal{P}(x_i^0|\mathbf{x}^T) = \sum_{\{x_j^0: j\neq i\}} \mathcal{P}(\mathbf{x}^0|\mathbf{x}^T)$ are computed, one simply ranks nodes in decreasing order of $\mathcal{P}(x_i^0 = I|\mathbf{x}^T)$.

6.2. BP Equations

For the sake of convenience, let us recall the general form of the BP equations. For a factorized probability measure on $\underline{z} = \{z_i\}$,

$$M(\underline{z}) = \frac{1}{Z}\prod_a F_a(\underline{z}_a) \tag{6.2.1}$$

where $\underline{z}_a$ is the subvector of variables that F_a depends on, the general form of the equations is

$$p_{F_a\to i}(z_i) = \frac{1}{Z_{ai}} \sum_{\{z_j : j\in\partial a\setminus i\}} F_a\left(\{z_i\}_{i\in\partial a}\right) \prod_{j\in\partial a\setminus i} m_{j\to F_a}(z_j) \tag{6.2.2}$$

$$m_{i\to F_a}(z_i) = \frac{1}{Z_{ia}} \prod_{b\in\partial i\setminus a} p_{F_b\to i}(z_i) \tag{6.2.3}$$

$$m_i(z_i) = \frac{1}{Z_i} \prod_{b\in\partial i} p_{F_b\to i}(z_i) \tag{6.2.4}$$

where F_a is a *factor* (i.e. ψ_i, ϕ_{ij}, γ_i, ζ_i^0, ζ_i^t or $\mathcal{G}_i$ in our case), z_i is a variable (i.e. (t_i, g_i), $(t_i^{(j)}, g_i^{(j)}, t_{ji})$, x_i^0 or x_i^T in our case), ∂a is the subset of indices of variables in factor F_a and ∂i is the subset of factors that depend on z_i. Eq. (6.2.4) for variable x_i^0 directly gives the posterior estimation of the probability of node i being in state I at time 0, i.e. being the patient-zero, and Eq. (6.2.4) for variable x_i^T gives the posterior estimation on the real state of individual i at time T.

It is possible to devise an efficient strategy for computing $p_{\psi_i\to j}\left(t_i^{(j)}, t_{ji}, g_i^{(j)}\right)$ in equation (6.2.2) for node factor ψ_i in (6.1.14), that can be computed in linear time in the degree of vertex i (Appendix B.1).

Factor ϕ_{ij} in Eq. (6.1.13) involves two (aggregated) variables: $(t_i^{(j)}, g_i^{(j)}, t_{ji})$ and $(t_j^{(i)}, g_j^{(i)}, t_{ij})$ (see Fig. 6.1.2). Variables $t_i^{(j)}, t_j^{(i)}, t_{ij}, t_{ji}$ have $T+2$ states, whereas variables g_i, g_j take values in $0, \ldots, G$. Given a distribution of recovery delays, it is sufficient to take G such that the weight of the tail of the distribution $\sum_{g=G}^{\infty} P(g)$ is small enough when compared to $1/N$. In the case of the geometric distribution of recovery delays, one can take e.g. $\sum_{g=G}^{\infty} P(g) = \sum_{g=G}^{\infty} p(1-p)^g = (1-p)^G \sim 1/N$, i.e. G needs to grows only logarithmically with N, and can of course be truncated at T. A naive implementation of the BP equations for factor ϕ_{ij} takes thus $O(T^4G^2)$ operations, that can be still too expensive in practical applications.

It is possibile to use a simpler representation for the messages, in which one just retains information on the relative timing between infection time $t_i^{(j)}$ for a node i and the infection propagation time t_{ji} on its link with node j, introducing the variables

$$\sigma_{ji} = 1 + \operatorname{sign}\left(t_{ji} - \left(t_i^{(j)} - 1\right)\right), \tag{6.2.5}$$

effectively reducing the complexity of messages from $O(T^2G)$ to $O(TG)$ real numbers. Appendix B.1 shows the computation of $p_{\phi_{ij}\to j}(t_j, \sigma_{ij}, g_j)$ from equation (6.2.2) for factor ϕ_{ij} with this simplification for the messages, bringing down the computation time from $O(T^4G^2)$ to $O(TG^2)$ operations. It also shows how to compute efficiently the BP equation for factor ψ_i corresponding to the simplified version of the messages $p_{\psi_i\to j}\left(t_i^{(j)}, \sigma_{ji}, g_i^{(j)}\right)$. The computation of all updates of each factor ψ_i takes $O(TG|\partial i|)$ operations, amounting to $O(TG|E|)$ operations for all ψ nodes per iteration. The update equations for the remaining factors, γ_i in Eq. (6.1.5), ζ_i^0, ζ_i^T in Eq. (6.1.10) and $\mathcal{G}_i$ in Eq. (6.1.1), can be computed in a straightforward way from (6.2.2), as they involve a very small number of variable states each.

The overall needed computation time of the update is dominated by the updates of ϕ_{ij} nodes, and amounts to $O(TG^2|E|)$ operations per iteration, where $|E|$ is the number of edges of the original contact graph. To give a rough estimation, in a C++ implementation, in a graph with 1000 nodes and 4000 contacts (edges), with $T = G = 20$, the computation takes about a minute per iteration on a single cpu. The full computation needs a few hundred iterations, i.e. about three hours on a

single cpu. Values of G smaller than T (within the estimation given above) can be used to reduce the computation time for larger T. Regarding memory usage, the BP equations with simplified messages need to store $O(TG|E|)$ real numbers.

6.3. Dealing with noisy observations

As I pointed out in the first section, it is often assumed that the state of every node is known at the observation time T with no uncertainty, that being the case also for the original work of Ref. [**15**] that introduced the BP method described in section 6.1. It is though realistic to assume that each observation carries some level of noise: every clinical test for determining the state of an individual is affected by some amount of error, and this possibility has to be taken into account in the inference method.

The major contribution of the work described in this chapter is the introduction of the general concept of *observation model* for the inference problem, which is suitable to represent the different cases of incomplete and noisy data using a common notation. Let us suppose, for simplicity, that noise acts independently on each site. If at time T node i is in the state x_i, it will have a certain finite probability of being observed in any of the other states. Let us introduce a new variable $y_i^T \in \{S, I, R\}$ for the observed state of node i, and assume in the following that the noise level is known (as it is the case for the majority of clinical tests). The simple modification of the original factor graph amounts in the introduction of an additional evidence term reflecting the probability of the observed state y_i^T given the true state x_i^T, set by the transition matrix $n_i\left(y_i^T|x_i^T\right)$.

In the modified factor graph, the new observed-state variables y_i^T are fixed to their value given by the experimental observation (by means of a delta function representing an infinite external field), while BP traces over the configurations of the true-state variables x_i^T. More explicitly, the modified factor graph shown in Fig. 6.1.3b shows a ζ_i^T factor node attached to the true-state variable x_i^T, which is linked to the observed state y_i^T (which is a constant) through the node $n_i\left(y_i^T|x_i^T\right)$. The posterior distribution now takes the form:

$$\mathcal{P}\left(\mathbf{x}^0|\mathbf{y}^T\right) \propto \sum_{\mathbf{x}^T,\mathbf{t},\mathbf{t}_{ij},\mathbf{g}} \mathcal{Q}'(\mathbf{x}^T,\mathbf{g},\mathbf{t},\mathbf{t}_{ij},x_0) \tag{6.3.1}$$

where

$$\mathcal{Q}'(\mathbf{x}^T,\mathbf{g},\mathbf{t},\mathbf{t}_{ij},x_0) = \frac{1}{Z}\prod_{i<j}\phi_{ij}\prod_i \psi_i \mathcal{G}_i\gamma_i\zeta_i n_i. \tag{6.3.2}$$

If one then introduces a map $\rho(s)$ from indices $i \in \{1,2,3\}$ into configurations of the x variables, such that $\rho(1) = S,\ \rho(2) = I,\ \rho(3) = R$, one can define the *observational transition matrix* (OTM) $O_{s,t}^{(i)}$ whose elements are the transition probabilities:

$$O_{s,t}^{(i)} = n_i\left(\rho(s),\rho(t)\right). \tag{6.3.3}$$

The case in which observations are complete and noiseless corresponds to an identity matrix $O_{s,t}^{(i)} = \delta_{st}$. In the following sections, I will cover some interesting examples of applications of this scheme to confused and noisy observations. Note that, in this generalized scheme, the case of partial observations corresponds to a totally uniform OTM $O_{s,t}^{(i)} \equiv \frac{1}{3}$ for unobserved nodes.

6.3.1. Inference of epidemic source from confused observations. In some situations, it could be hard to distinguish between nodes that already recovered from a disease and nodes that did not contract it at all [**73**]. Let us then allow only two types of observed states $x_i^T \in \{I, N\}$, where N stands for Not-Infected: in this setting Susceptible and Recovered nodes are not distinguishable. This corresponds to choosing the following OTM:

$$O^{(i)} = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}.$$

Firstly, I verified the performances of the BP algorithm on a uniform setting provided by random regular graphs with identical infection parameters (λ, μ) for all nodes and links. All epidemic propagations were initiated from a unique source (the *patient zero*, or *seed*). For each node, the BP algorithm provides an estimate of the posterior probability that the node got infected at a certain time, and thus also the probability that the node was the origin of the epidemic. Nodes are then ranked in decreasing order with respect to the estimated probability of being the origin of the observed epidemic: the position of the true origin in the ranking provided by the algorithm is a good measure of the efficacy of the method. In what follows, the ranking of the true origin of the epidemic is indicated with i_0, and $|G|$ is the number of nodes in the graph G.

Figure 6.3.1 displays the absolute rank of the true infected site i_0, for a set of $M = 1000$ simulated epidemic propagations with $\lambda = 0.6$ and $\mu = 1$ on random regular graphs of size $N = 1000$ and degree $d = 4$ ($T = 10$). The probability of perfect inference of the patient-zero is also reported. The quantities of interest are plotted as functions of the normalized epidemic size $N_{IR} = \frac{|I|+|R|}{|G|}$ (i.e. the fraction of infected or recovered sites), whose values are discretized with intervals of width equal to 0.05. In all the following figures, the rare cases with very low epidemic size ($N_{IR} < 0.3$ in Fig. 5, $N_{IR} < 0.2$ elsewhere) have been discarded, since inference is practically infeasible when the number of infected nodes is extremely low.

For each set of data, the symbols report the mean value obtained averaging over the samples belonging to that interval and the error bars indicate the corresponding standard deviation. The average fraction of Infected nodes and the fraction of samples in each bin are reported as a reference. The algorithm is very effective in identifying the patient-zero for all values of the normalized epidemic size.

The marginals of the infection time for each node provide a method to "correct" observations, that can be exemplified as follows, considering the problem of discriminating between Susceptible and Recovered nodes. *Receiver Operating Characteristic* (ROC) curve, namely a plot of the 'true positive rate' against the 'false positive rate', is a method for quantifying the accuracy of such a binary classification problem. Constructing the ROC curve in the present case is very easy: after ranking the N nodes on the base of their marginal $\mathcal{P}\left(t_i = \infty | \mathbf{x}^0\right)$, one takes one step upward in the ROC whenever a true positive case is encountered ($y_i^T = x_i^T = S$) or one step rightward in case of a false positive ($y_i^T \neq x_i^T$). I performed this discrimination analysis for each sample and then computed the average value of the area under the ROC curve, that gives indication of the fraction of correctly classified nodes. Figure 6.3.1 also shows the average ROC area, which reveals that the inference algorithm allows a very good discrimination between S and R nodes.

The same analysis for a random graph with power-law degree distribution, obtained using the Barabasi-Albert model [**84**], is reported in Fig.6.3.2. When the observation time T is sufficiently small ($T = 7$ in Fig.6.3.2), the performance of the algorithm is high. When longer observation times are considered, epidemics tend to cover the whole network and convergence issues emerge. In this

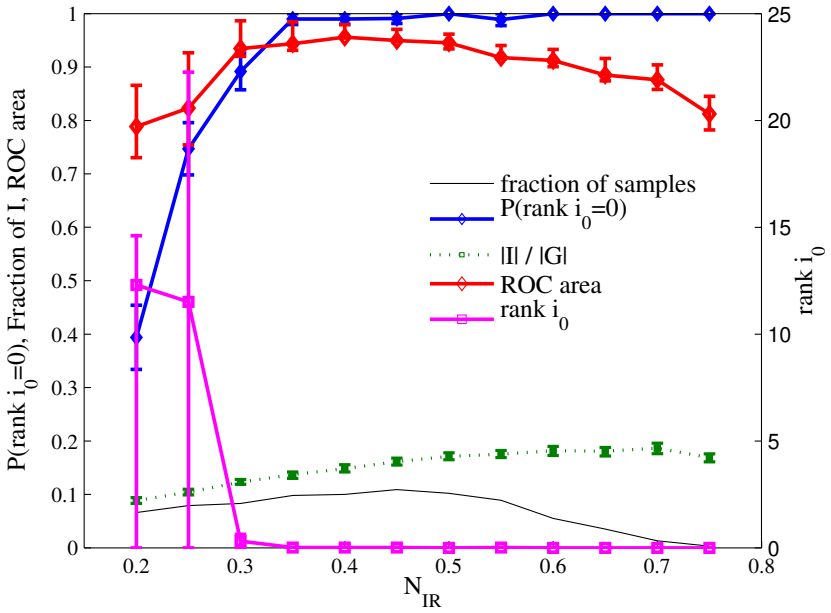

FIGURE 6.3.1. Probability of perfect inference of the patient-zero (blue solid line), average absolute rank of the true patient-zero (violet solid line), average ROC area (red solid line) and average fraction of Infected nodes $\frac{|I|}{|G|}$ (green dotted line) as a function of the rescaled epidemic size $\frac{|I|+|R|}{|G|}$. The fraction of the M samples belonging to each bin of the rescaled epidemic size is also indicated. The realization of the epidemic process is propagated for $T = 10$ steps with $\lambda = 0.6$ and $\mu = 1$. Observations are confused, i.e. $x_i^t \in \{I, N\}$. Simulations were run over $M = 1000$ samples of random regular graphs with $N = 1000$ nodes and degree $d = 4$.

regime, most of the infected nodes have already recovered at the observation time T (and thus they cannot be distinguished from the susceptible ones anymore). This causes a rapid decay of the available information content, that explains the performance degradation. A similar effect arises also on random regular graphs, but at longer times, as we will see in Section 6.3.2.

In summary, even when supplied with confused observations, BP shows striking ability to discriminate between Recovered and Susceptible nodes, provided that there is enough information at the chosen observation time T.

6.3.2. Inference of the epidemic source with noisy observations. As a simple model for observational noise, one may consider a totally symmetric setting where a node with a state x has probability $1-\nu$ of being correctly observed in state x, and probability ν of being observed incorrectly in one of the two remaining states, distributed uniformly among the two. For example, node i could be I (Infected) at the observation time T, and, for a given noise level ν, there will be an equal probability $\frac{\nu}{2}$ for node i to be observed in the R (Recovered) or S (Susceptible) state. This setting corresponds

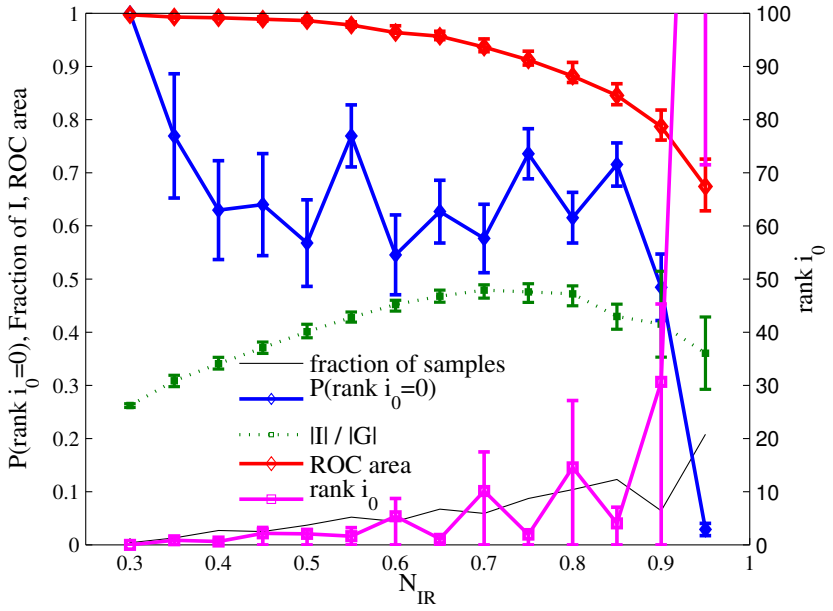

FIGURE 6.3.2. Probability of perfect inference of the patient-zero (blue solid line), average absolute rank of the true patient-zero (violet solid line), average ROC area (red solid line) and average fraction of Infected nodes $\frac{|I|}{|G|}$ (green dotted line) as a function of the rescaled epidemic size $\frac{|I|+|R|}{|G|}$. The fraction of the M samples belonging to each bin of the rescaled epidemic size is also indicated. The realization of the epidemic process is propagated up to the time step $T = 7$ with $\lambda = 0.6$ and $\mu = 0.5$. Observations are confused, i.e. $x_i^t \in \{I, N\}$. Simulations were run over $M = 1000$ samples of Barabasi-Albert graphs with $N = 1000$ nodes and average degree $\hat{d} = 4$.

to the following OTM:

$$O^{(i)} = \begin{pmatrix} 1-\nu & \frac{\nu}{2} & \frac{\nu}{2} \\ \frac{\nu}{2} & 1-\nu & \frac{\nu}{2} \\ \frac{\nu}{2} & \frac{\nu}{2} & 1-\nu \end{pmatrix}. \tag{6.3.4}$$

Fig. 6.3.3 displays the average rank of the true origin of the epidemics (left) and the probability of inferring the true patient-zero (right) for various levels of the observational noise up to $\nu = 0.4$, in a set of $M = 1000$ single-source epidemic propagations with $\lambda = 0.6$ and $\mu = 1$ on random regular graphs with $N = 1000$ nodes and degree $d = 4$. . The low values of the average rank obtained demonstrate that the BP algorithm is able to perform extremely well up to very high levels of noise. The corresponding ROC curves are plotted in Fig. 6.3.4. In Fig. 6.3.5 I show a representative situation in random graphs (the picture is similar in scale-free graphs, as we argued in Section6.3.1) where the observation time is systematically varied at various level of noise. It turns out that the ratio of infected nodes to epidemic size is critical for inference: when observation time is too long so that the majority of infected individuals have recovered, it is much more difficult to find the patient-zero in the noisy and

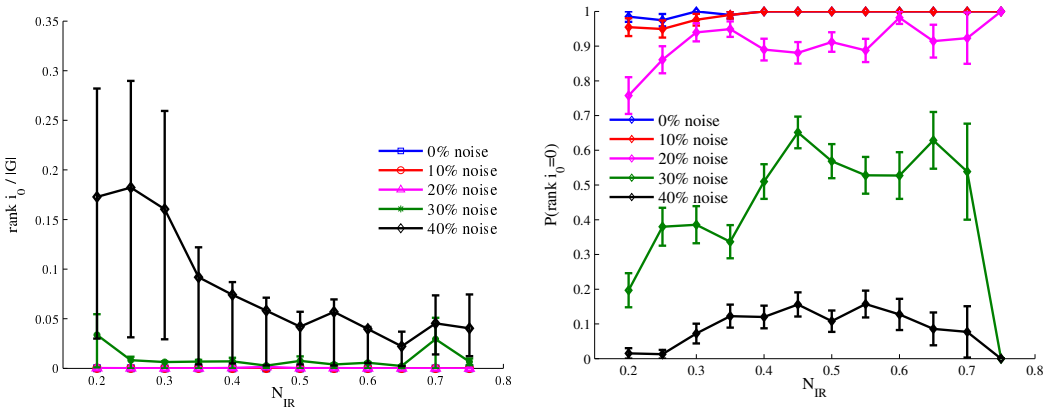

FIGURE 6.3.3. Average normalized rank of the true patient-zero (left) and probability of ranking it in the first place (right) as a function of epidemic size $\frac{|I|+|R|}{|G|}$ for various levels of noise ν in the observation (the error-bars indicate the standard deviation computed on the sub-sample corresponding to a given epidemic size). Each curve refers to $M = 1000$ samples of Random Regular graphs with $N = 1000$ nodes and degree $d = 4$. Epidemics is propagated until $T = 10$ with $\lambda = 0.6$ and $\mu = 1$.

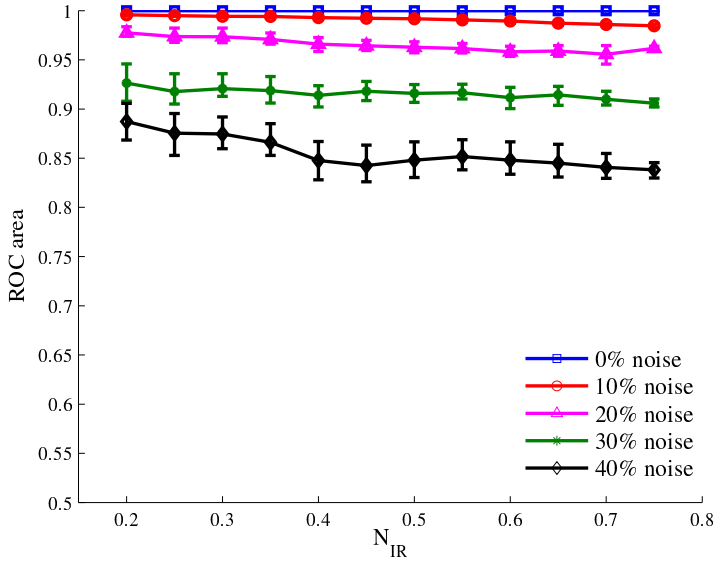

FIGURE 6.3.4. Average area of the ROC curve of the inference of the state of noisy variables as a function of epidemic size $\frac{|I|+|R|}{|G|}$ for various levels of noise ν in the observation (the error-bars indicate the standard deviation computed on the sub-sample corresponding to a given epidemic size). Each curve refers to $M = 1000$ samples of Random Regular graphs with $N = 1000$ nodes and degree $d = 4$. Epidemics is propagated until $T = 10$ with $\lambda = 0.6$ and $\mu = 1$.

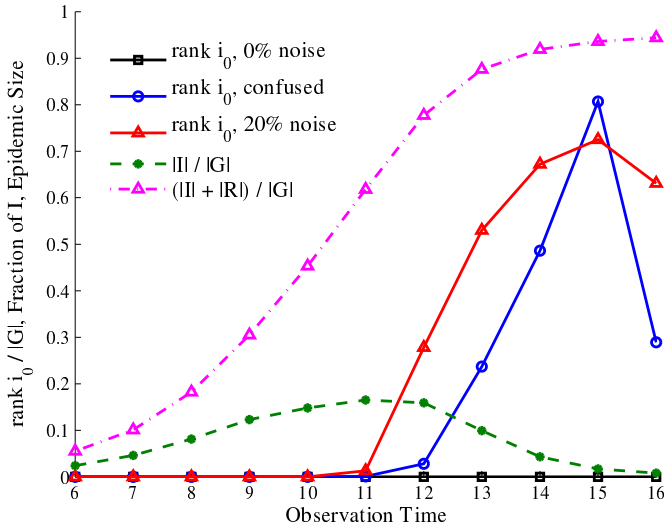

FIGURE 6.3.5. Normalized rank of the true patient-zero (solid lines), fraction of Infected nodes $\frac{|I|}{|G|}$ (green dotted line) and rescaled epidemic size $\frac{|I|+|R|}{|G|}$ (dotted purple line) as a function of observation time T for a single realization of the epidemic process, propagated with $\lambda = 0.6$ and $\mu = 1$, on a random regular graph with $N = 1000$ nodes and degree $d = 4$. Observations are complete (black solid line, superimposed to x axes), confused (blue solid line), and with noise 20% noise level (red solid line). Values of the normalized rank greater than 0.5 are meaningless: they are the realization of a random variable with average close to 0.5, and they are evidence that for large T the inference algorithm is unable to identify the patient-zero with better precision than pure chance.

confused case. As it can be seen in the figure, this sharp change of behaviour (manifested at $T = 11$) is present even in single instances.

6.4. Inference of Epidemic Parameters

In what has been discussed so far, the epidemic parameters have been considered as additional data in the problem at hand: it is certainly reasonable to assume that, for certain types of diseases, the average rates of infection and recovery are known to a certain extent, and some information is available regarding at least their range of variation. The additional problem of inferring the parameters can be cast in a Bayesian framework with the introduction of the log-likelihood of the epidemic parameters for the observation $\mathbf{x}^T$. The log-likelihood of the parameters $-f(\lambda, \mu) = \log Z(\lambda, \mu)$ equals $\log \mathcal{P}\left(\mathbf{x}^T|\lambda, \mu\right)$ and the latter can be computer by means of a sum over all the possible paths and the initial conditions with a fixed observation:

$$\mathcal{P}\left(\mathbf{x}^T|\lambda, \mu\right) = \sum_{\mathbf{t},\mathbf{g},\mathbf{x}^0} \mathcal{P}\left(\mathbf{x}^T|\mathbf{t}, \mathbf{g}\right) \mathcal{P}\left(\mathbf{t}, \mathbf{g}|\mathbf{x}^0\right) \mathcal{P}\left(\mathbf{x}^0\right) = Z\left(\lambda, \mu\right). \tag{6.4.1}$$

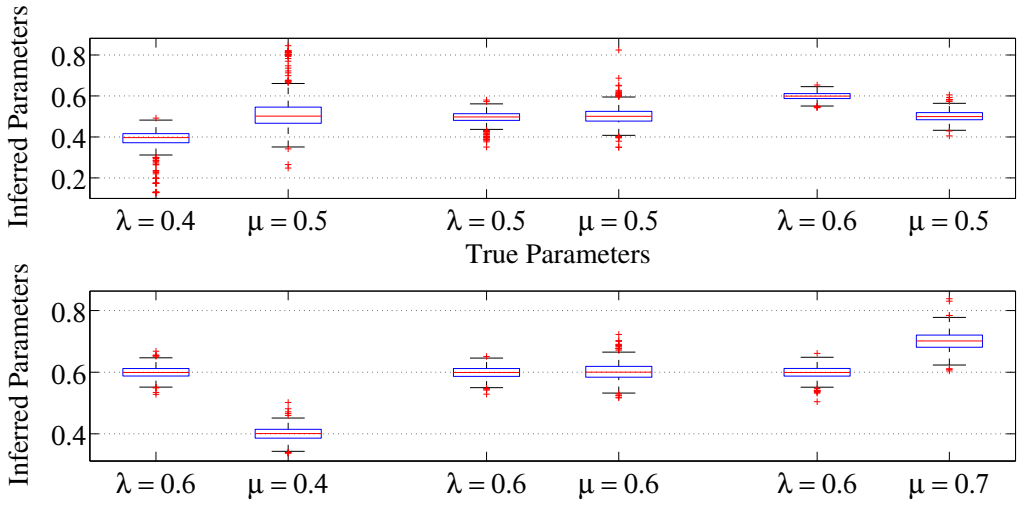

FIGURE 6.4.1. Inferred epidemic parameters for six different configurations of true (λ, μ) parameters. Forward epidemic is simulated until observation time $T = 10$. Each pair of boxes refers to $M = 1000$ instances of Random Regular graphs with $N = 1000$ nodes and degree $g = 4$. Box edges signal the 25th and 75th percentiles, the central red lines is the median. Whiskers extend to cover 99.3% of the data for a gaussian distribution. Outliers are marked as red points outside the whiskers.

In the Bethe approximation, the log-likelihood is just the free energy of the system, and can be expressed as a sum of local terms depending on the BP messages:

$$-f = \sum_a f_a + \sum_i f_i - \sum_{(ia)} f_{(ia)} \tag{6.4.2}$$

where

$$f_a = \log\left(\sum_{\{z_i : i\in\partial a\}} F_a\left(\{z_i\}_{i\in\partial a}\right) \prod_{i\in\partial a} m_{i\to a}(z_i)\right) \tag{6.4.3}$$

$$f_{(ia)} = \log\left(\sum_{z_i} m_{i\to a}(z_i) p_{F_a\to i}(z_i)\right) \tag{6.4.4}$$

$$f_i = \log\left(\sum_{z_i} \prod_{b\in\partial i} p_{F_b\to i}(z_i)\right) \tag{6.4.5}$$

The most naive way of performing a log-likelihood maximization in the space of parameters would be an exhaustive search. This was initially done in Ref. [**15**]. I worked on a generalization of this procedure that would allow to infer both the patient-zero and the epidemic parameters at the same time. The method discussed in this section resides on the simplifying assumption of homogeneity of the epidemic parameters $\lambda_{ij} = \lambda$ and $\mu_i = \mu$. As I will show in chapter 8, the same method can be used in the more general framework of network reconstruction.

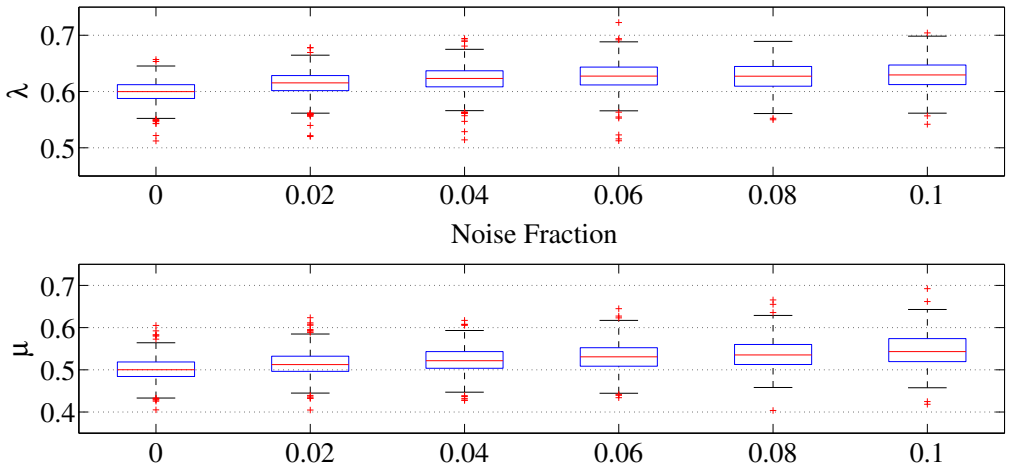

FIGURE 6.4.2. Inferred epidemic parameters for different observational noise rates ν. Forward epidemic is simulated until observation time $T = 10$. Each box refers to $M = 1000$ instances of Random Regular graphs with $N = 1000$ nodes and degree $g = 4$. Box edges signal the 25th and 75th percentiles, the central red lines is the median. Whiskers extend up to cover 99.3% of the data for a gaussian distribution. Outliers are marked as red points outside the whiskers.

Once the free energy is expressed as a function of the parameters, its minimum (maximum of the log-likelihood) may be searched with a simple gradient descent procedure, by means of the following updates:

$$\lambda \leftarrow \lambda - \epsilon \frac{\partial f}{\partial \lambda} \tag{6.4.6}$$

$$\mu \leftarrow \mu - \epsilon \frac{\partial f}{\partial \mu} \tag{6.4.7}$$

with ϵ a free convergence parameter (note that the minus sign comes from the definition of the free energy). A detailed derivation of the expression of the derivatives $\frac{\partial f}{\partial \lambda}$ and $\frac{\partial f}{\partial \mu}$ of the Bethe free energy is reported in Appendix B.2. In principle, the expressions obtained using the Bethe free energy are valid only at the BP fixed point, and one should let BP updates converge before making a step of gradient descent. In practice, it is sufficient to interleave BP and gradient descent updates in order to obtain equivalent results. In order to validate the method, I performed extensive simulations with a wide range of parameters and found that, for reasonable fraction of infected nodes at the observation time, this method simultaneously identifies the patient-zero perfectly and finds good estimates of the epidemic parameters. Some examples of inferred parameters are shown in Fig. 6.4.1 for six different configurations of (λ, μ) parameters, with each pair of box plots referring to $M = 1000$ samples.

The inference of parameters can be performed also in the presence of observational noise. Fig. 6.4.2 shows an example of inference for increasing levels of noise in the observation, as defined in section

6.3. Also in this case the patient-zero is detected with probability 1 and the inferred parameters are good estimators of the true values, even up to a significant fraction of noise.

In order to quantify the performance of the new method, I built a simple comparison with a very simple procedure. Let us suppose that, given a graph, the distribution of number of infected $I(\lambda,\mu)$ and recovered $R(\lambda,\mu)$ individuals is known for each value of λ and μ. When confronted with a realization of the epidemic process, one may choose those λ,μ whose statistical features are closest (in some sense to be defined) to the one observed. In practice, I run 1000 random epidemics for each combination of values $\lambda \in \{0.05, 0.1, \ldots, 0.95\}$, $\mu \in \{0, 0.05, \ldots, 1\}$ and computed the mean of the number of infected $I_{mean}(\lambda,\mu)$ and recovered $R_{mean}(\lambda,\mu)$ individuals. Given an observation with I

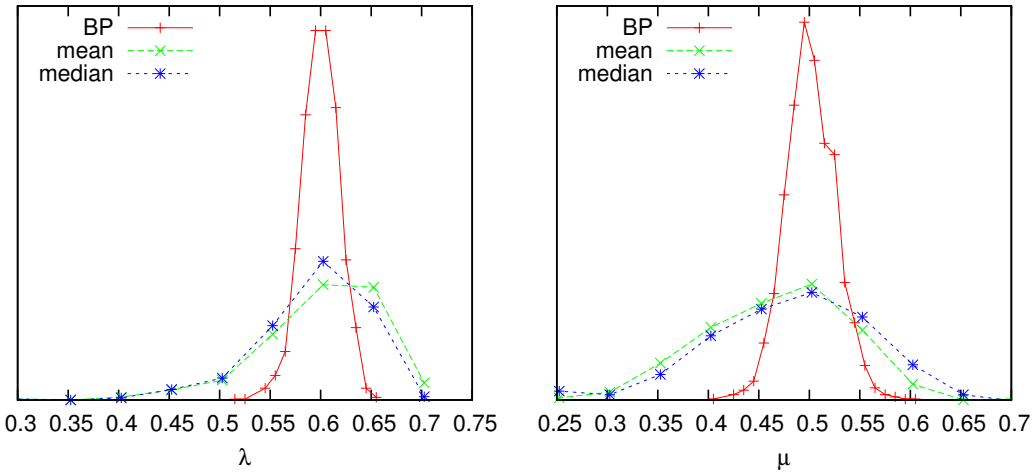

FIGURE 6.4.3. Comparison of inference of epidemic parameters for 200 random realizations with $\lambda = 0.6$, $\mu = 0.5$ between BP and the naive method consisting in finding the couple (λ^*, μ^*) which is closest in terms of mean (resp. median) number of infected and recovered individuals in euclidean distance. The distributions for the inference with BP correspond to the fifth example reported in Fig.6.4.1 and the first in Fig.6.4.2.

infected and R recovered individuals, the result of the inference is simply

$$(\lambda^*, \mu^*) = \arg\min_{\lambda,\mu} (I - I_{mean}(\lambda,\mu))^2 + (R - R_{mean}(\lambda,\mu))^2.$$

Fig. 6.4.3 shows the distributions of λ^* and μ^* found by the above procedure based on 200 epidemic realizations with $\lambda = 0.6$ and $\mu = 0.5$, along by the same distribution as found by the interleaved BP gradient ascent of the likelihood function. The results show that the BP-based procedure is able to infer the correct parameter $\lambda = 0.6$ and $\mu = 0.5$ with much higher accuracy. The same procedure using the *median* instead of the mean (and computing thus I_{median} and R_{median}) yields very similar results.

6.5. Summary

In this Chapter, I discussed the problem of inferring the origin of an epidemic propagation on a network from a single snapshot of its collective state, and described a generalization of a previously developed inference scheme to the more realistic scenario of noisy observations. As I pointed out in Section 6.3.2, the effectiveness of the proposed inference strategy is constrained by the amount of information available in the snaphost, which is generally a function of the delay time and the noise level.

Belief Propagation performs well even when observations are uncertain or completely confused, such as the case where one is unable to distinguish between observed states. When coupled to a gradient ascent procedure, BP equations provide a variational strategy for inferring epidemic parameters at the same time. The method described in this chapter has been presented in Ref. [**85**].

In the presence of multiple epidemic cascades on a given graph, the present approach can be extended to infer the infection probabilities of any putative link, providing an efficient method for reconstructing the entire network: this will be the subject of chapter 8. In the next chapter, I will discuss a further generalization of our inference method which is capable of dealing with real contact data in continuous time, without resorting to a time discretization.

CHAPTER 7

Inverse dynamics in continuous-time contact networks

The inference method discussed in chapter 6 considered a discrete time stochastic SIR model on a network: this means that, whenever precise temporal information is available about the contacts between individuals, single contacts have to be binned in order to construct an effective network with (in principle time dependent) infection probabilities λ_{ij} on each link. This is also the case for any known inference algorithm so far in the literature. For a specific continuous-time epidemic process, an optimal estimator on general trees was put forward by Pinto et al. [**69**], but it is of limited use on general graphs, and lacks a Bayesian foundation.

Most traced real networks [**86, 87, 88**] consist however of a list of timed quasi-instantaneous interaction events (resolution is typically down to time intervals of the order of seconds). These contact networks have been inaccessible for decades, but thanks to recent advances in technology miniaturization (e.g. by means of RFID-endowed badges to signal the proximity between individuals) and the popularization of the Internet (e.g. for the construction of databases of self-reported interactions), at least in simple and controlled scenarios the interaction patterns of individual contacts can be almost entirely reconstructed [**88, 86**]. Modern computational epidemiology can thus rely on accurate data and on powerful computers to run large-scale simulations of stochastic compartment models on real contact networks [**89, 90**].

Here I will discuss a recent development which was done in collaboration with Alfredo Braunstein, namely a continuous-time inference framework which is able to deal with datasets without any discretization procedure [**91**]. In what follows, I will describe a continuous-time infection model, where contacts are assumed to be instantaneous events during which the disease can be passed. The development of the method is conceptually very similar to what has been introduced in the previous chapter, in that it relies on the construction of a static graphical model representation for the infection dynamics (section 7.1), on top of which an approximate inference method based on Belief Propagation is employed to compute in an efficient way the marginals over the posterior distribution of initial spreaders (section 7.2). In section 7.3 the method is tested on two datasets of real contact networks, and in section 7.4 its performance is compared to the corresponding discretized version of the model on one of the two datasets.

7.1. Graphical model representation

Let us consider an evolving contact network composed of N nodes, and define the model as follows: let time $t \in \mathbb{R}_+$ be continuous, and contagion events be instantaneous with probability λ. Let us denote by $t_i \in \mathbb{R}_+$ the infection time of node i. Each pair ij will be in contact in a discrete set of instants $T_{ij}(0) < T_{ij}(1) < \cdots < T_{ij}(n_{ij})$ (ideally, given by the real network trace dataset). Let us also assume $T_{ij}(r) = \infty$ for $r > n_{ij}$. As time t advances, contagion will happen with probability λ if i is infected, j is susceptible and $\exists r$ such that $T_{ij}(r) = t$. Recovery of individuals will happen at a time $t_i + g_i$ with a given recovery probability distribution $G_i(g_i)$.

Define $r_{ij} \in \mathbb{N}_0$ for $ij \in E$ a set of i.i.d. variables with geometric distribution $R(r_{ij}) = \lambda(1-\lambda)^{r_{ij}-1}$. Suppose the recovery time g_i of node i is distributed with continuous distribution G. Given $\{g_k\}$ and $\{r_{ki}\}$, infection time t_i satisfy deterministically:

$$t_i = \min_{k\in\partial i: T(r_{ki}+\min\{r:T_{ki}(r)>t_k\})<t_k+g_k} T_{ki}(r_{ki}+\min\{r : T_{ki}(r) > t_k\}), \tag{7.1.1}$$

because neighbor k will be infected in the time interval $[t_k, t_k + g_k]$ and will possibly transmit the disease at some time $T_{ki}(r)$. Let us define

$$t_{ki} = \begin{cases} T_{ki}(r_{ki}+\min\{r : T_{ki}(r) > t_k\}) & \text{if } T_{ki}(r_{ki}+\min\{r : T_{ki}(r) > t_k\}) < t_k + g_k \\ \infty & \text{else} \end{cases} \tag{7.1.2}$$

the "delayed" infection time. Then

$$t_i = \min_{k\in\partial i} t_{ki} \tag{7.1.3}$$

Note that t_i will take values in $h_0 < \cdots < h_n$ the set of all incoming contact times $\cup_{k\in\partial i, r=0,\ldots n_{ki}} T_{ki}(r)$. In terms of time indices s_{ij}, such that $t_{ij} = T_{ij}(s_{ij})$, we have $s_{ij} = S_{ij}(\min_{k\in\partial i} T_{ki}(s_{ki}), r_{ij}, g_i)$ where

$$S_{ij}(t_i, r_{ij}, g_i) \stackrel{def}{=} \begin{cases} r_{ij}+\min\{r : T_{ij}(r) > t_i\} & \text{if } T_{ij}(r_{ij}+\min\{r : T_{ij}(r) > t_i\}) < t_i + g_i \\ \infty & \text{else} \end{cases} \tag{7.1.4}$$

In analogy to section 6.1, it is now possibile to write down the Boltzmann distribution of dynamical trajectories in the following forms:

$$\mathbf{P}(\mathbf{t}, \mathbf{s}, \mathbf{r}, \mathbf{g}) \propto \prod_i G(g_i)\, \delta\left(t_i, \min_{k\in\partial i} T_{ki}(s_{ki})\right) \prod_{ij} R(r_{ij})\, \delta(s_{ij}, S_{ij}(t_i, \{r_{ki}\}, g_i)) \tag{7.1.5}$$

7.2. Belief Propagation Equations

The main observation behind the construction of an inference framework by means of a graphical model representation is that, although the spreading process in defined in continuous time, the infection times are constrained to be in the finite discrete set $\{T_{ij}(k)\}_{k=1\ldots n_{ij}}$. Once a conditional probability distribution over all the possibile trajectories is defined, and the priors over the initial states are inserted as in Eq. (6.1.5), the Bayesian approach is defined in exactly the same way as explained in sections (6.1) and (6.2).

Once again, in order to perform inference, one needs to sum over all the dynamical trajectories which are compatible with the observation, a problem that will be here tackled with the use of Belief Propagation. In what follows, the observation is assumed to happen immediately after the last contact, but other choices may be included in a straightforward manner.

By defining factors

$$\psi_i\left(t_i, \{s_{ki}, s_{ik}\}_{k\in\partial i}, \{r_{ik}\}_{k\in\partial i}\right) = G(g_i)\, R(r_{ij}) \prod_{j\in\partial i} \delta(s_{ij}, S_{ij}(\{s_{ki}\}, \{r_{ki}\}, g_i))\, \delta\left(t_i, \min_{k\in\partial i} T_{ki}(s_{ki}), \right) \tag{7.2.1}$$

equation 7.1.5 can be simply recast as:

$$\mathbf{P} \propto \prod_i \psi_i \tag{7.2.2}$$

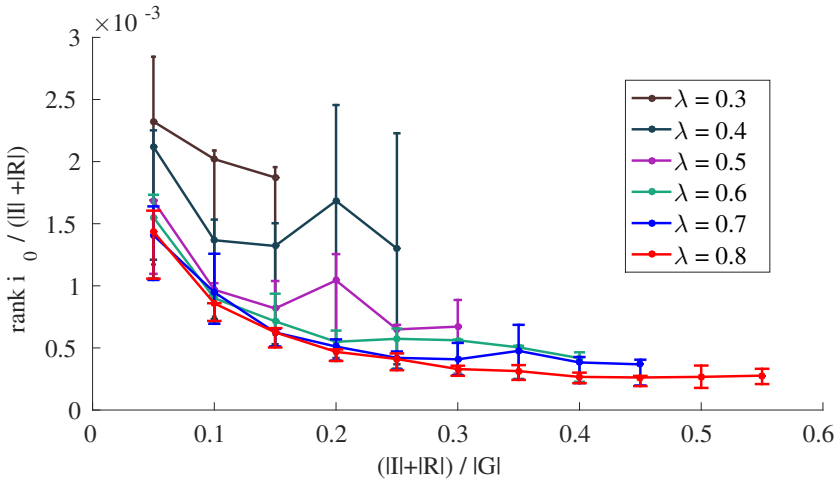

FIGURE 7.3.1. Average normalized rank $\frac{r_0}{|I|+|R|}$ of the true patient zero as a function of $N_{IR} = \frac{|I|+|R|}{|G|}$, the normalized epidemic size, for $\mu = 0.5/year$ and increasing values of the infection probability λ in the network of sexual contacts. Each curve represents a sample of $M = 1000$ random instances.

where a simple factorization over sites is apparent. There are only two types of BP message, namely $P_{ij}(s_{ij}, s_{ji})$ and $P_i(t_i)$, and the corresponding update equations are easily derived from 6.2.2 and 6.2.3. An efficient implementation of these equations together with estimates of the computational complexity of each update is shown in Appendix C. Once Belief Propagation converges, and similarly to the Bayesian approach in the discrete-time case, eq. (6.2.4) is used to compute the marginal probability $p_i(t_i = 0)$, which brings a posterior estimation of the probability for the node i to be active before the first contact. Sites are then ranked with respect to their posterior probability.

In complete analogy with section 6.3, it is not difficult to extend the present model to account for observations affected by some kind of uncertainty. One equivalently introduces the *Observational Transition Matrix* $n_i\left(y_k^i|x_l^i\right)$, containing the transition probabilities from the true state x_l^i to the observed state: in order to perform inference, the observed-state variables y^i are fixed with an infinite external field into their state, and the true-state variables x^i are traced over in the compatibility function ζ_i. The identity matrix $n_i\left(y_k^i|x_l^i\right) = \delta_{kl}$ corresponds to the a case in which no noise enters the observations.

7.3. Results on real networks

The new inference method was tested on two large evolving networks: a database of time-stamped sexual interactions and a network of face-to-face contacts in a high school.

The first dataset comes from a database of sexual encounters between clients and escorts on a Brazilian website, covering the beginning of the community, September 2002 through October 2008, and composed of a total of $E = 50185$ contacts between between 6642 escorts and 10106 sex-buyers. This kind of data are particularly relevant in the study of spreading of sexually transmitted infections (STI) and have been previously used to model the diffusion of HIV by means of simple SI-SIR compartmental models [**87**].

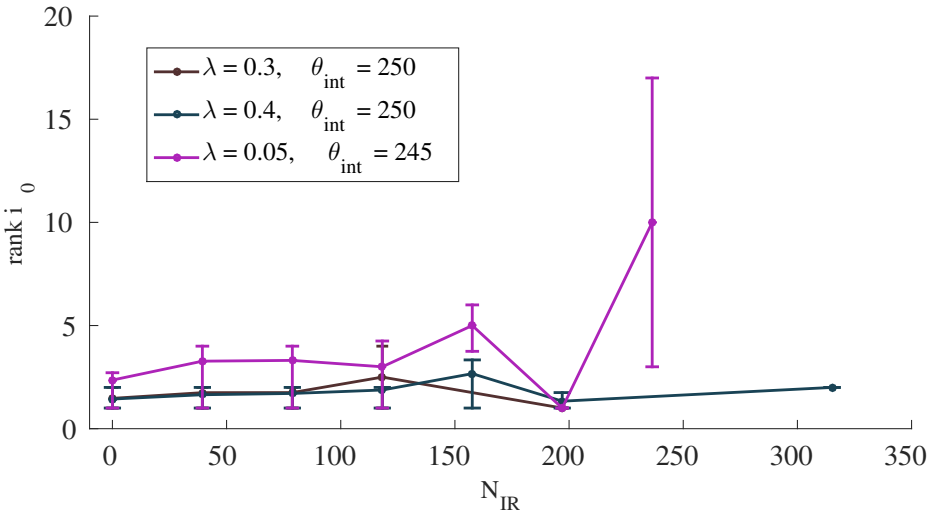

FIGURE 7.3.2. Average absolute rank i_0 as a function of $|I| + |R|$, the total epidemic size, in the network of face-to-face contacts in a high school for different values of the threshold θ_{int}. Each curve is an average over $M = 300$ ($\theta_{int} = 250$) or $M = 1000$ ($\theta_{int} = 245$) random instances. The infection probability λ is scaled accordingly in order to obtain epidemics with average dimension N_{IR}.

I built a bipartite evolving network, focusing on the last two years of steady state operation of the website ($E = 29628$ contacts, slightly over half of the dataset) in order to skip the initial period where encounters are very sparse. For each value of λ, I simulated $M = 1000$ single source epidemic propagations, with a recovery rate equal to $\mu = 0.5/year$. In Fig. 7.3.1 I show the average rank of the true first infected individual i_0 divided by the epidemic size $|I| + |R|$ as a function of the normalized epidemic size $N_{IR} = \frac{|I|+|R|}{|G|}$ (i.e. the fraction of infected and recovered sites), whose values are discretized in intervals of width equal to 0.1.

The second dataset consists of a collection of close proximity interactions (CPIs) obtained by means of wireless sensor network technology (TelosB motes) [**92**]. Data were collected in a US high school and provide an almost complete account of face-to-face interactions during a whole day at school. All in all 798 individuals were monitored, corresponding to the 94% of the total school community, and 2148991 unique close proximity records (CPR) were acquired. A single CPR consists of a close proximity detection event between two motes (max. 3 meters). The authors of the study perform an aggregation of the raw data in *interactions*, defined as continuous sequences of CPR between the same two nodes. My choice was to go back to the raw data and investigate the spreading process at the level of single CPR, using the intensity signal as a proxy for the closeness of a face-to-face contact (a detailed account is presented in [**92**], SI). I constructed a set of evolving networks by setting a threshold θ_{int} for the signal intensity of the motes, thus considering denser and denser networks as θ_{int} is decreased, and weaker and weaker (and distant) contacts are taken into account. Three representative examples are show in Fig. 7.3.2, which displays the average rank of the true first infected individual i_0 for different values of λ and threshold θ_{int}. Note that the infection probability λ has to be scaled down when θ_{int} is decreased in order to obtain epidemics with the same dimension distribution.

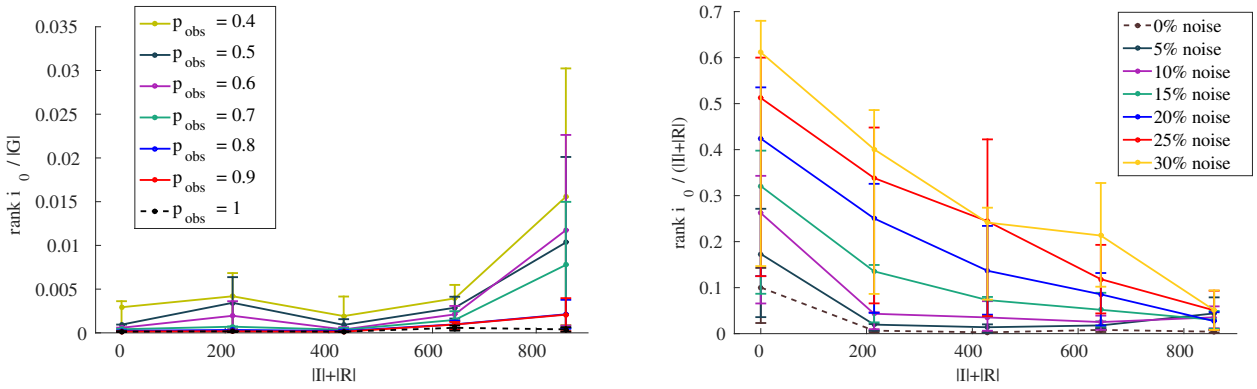

FIGURE 7.3.3. Inference performance with partial or noisy observations in the network of sexual contacts. In each curve we show the average rank absolute rank over $M = 100$ random instances of the true patient zero i_0 as a function of the total epidemic size $|I| + |R|$ for$\lambda = 0.2$, $\mu = 0.5$. Left panel: rank i_0 normalized over $|G|$ vs $|I| + |R|$ for different values of probability of observation p_{obs}; broken curve is the case with full observations. Right panel: normalized rank i_0 vs $|I| + |R|$ for different noise intensities ν; broken curve is the case with no noise.

7.3.1. Partial and noisy observations. In what follows, I will use the sexual encounters dataset, mostly because of its epidemiological relevance. Let us first consider the case of partial observations, i.e. the case in which only a subset of nodes are accessible for observation at time T: this is the standard realistic scenario in practical applications, when a complete monitoring of a full network is infeasible in the general case. I simulated a number of epidemic spreading in the contact network and modeled the partial observability by a fixed probability p_{ob} of observing a node at time T. Note that, in this generalized scheme, partial observability can be represented with a totally flat OTM $n_i\left(o'^i_k|o^i_l\right) = \frac{1}{3}$ for unobserved nodes. Results for decreasing values of p_{ob} in the network of sexual contacts are shown in the left panel of Fig. 7.3.3 (the complete observation case $p_{ob} = 1$ is shown in the dashed line for reference).

Turning to the problem of uncertain observations, I used the very same type of symmetric model for observational noise introduced in 6.3, described by the OTM in eq. 6.3.4. The right panel of Fig. 7.3.3 shows how the BP algorithm is highly robust even to a significant amount of noise, up to 30%.

7.4. Limitations of time-discretization

The standard practice when one deals with evolving networks is the construction of an effective network where the individual contacts are summed up and aggreggated in a link dependent infection probability $\lambda_{ij}(t)$, with t in some small discrete set. This time discretization, which obviously leads to a detriment both in the forward simulation and in inverse reconstruction, is essential because most inference paradigms rely on a purely static network, where infection probability over links can possibly vary in different (discrete) time-steps.

In the styudy that motivated the generalization discussed in chapter 6, the authors used the discretization procedure to perform the inference of the patient zero in the network of sexual contacts

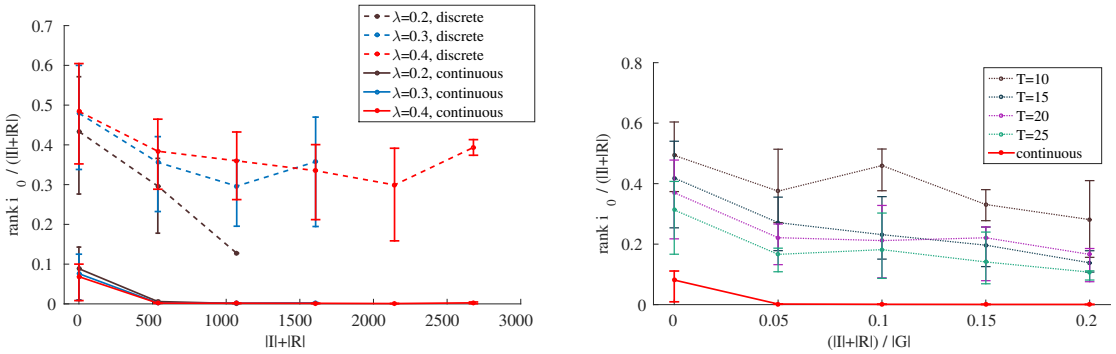

FIGURE 7.4.1. Left: Comparison between the continuous time method and the discretized version in the network of sexual contacts for different values of the infection probability λ and $\mu = 0.5/year$. Each curve is the average normalized rank $\frac{r_0}{|I|+|R|}$ over $M = 500$ samples of the patient zero as a function of epidemic size $|I| + |R|$. Full curves: continuous time version. Broken curves: contacts have been aggregated in effective contacts so that final time of observation is $T = 10$. Right: A comparison of the continuous time method versus the discretized version for $\lambda = 0.4$ and $\mu = 0.5/year$ with an increasing number of time bins.

after aggregating the time-stamped contacts in effective interactions of size ΔT. I performed similar numerical experiments, with the caveat that the forward simulations of the epidemic process were performed directly at the level of real time contacts, before the aggregation procedure. I then used the new continuous-time method on the same instances, in order to build a direct comparison. The difference in performance is striking at all values of λ, as can be seen in Fig. 7.4.1.

7.5. Summary

In this chapter, I showed how the Belief Propagation approach introduced in chapter 6 can be generalized to the case of interactions happening in continuous time. This makes possibile to use data set of real time interactions between individuals to uncover the paths that led to the spreading of a disease in an evolving network. The importance of this method is not purely methodological: in the epidemiological context, it is appealing to consider the possibility of applying this technique when all the interactions in a hospital or a relatively closed community are constantly monitored. Because of the generality of SIR as a model of spreading of activation over graphs, this method is in principle applicable in a variety of different situations, for example in the identification of the source of the initial spreader of a computer virus in a network.

CHAPTER 8

Inference of contact networks from sparse observation of cascades

The inference methods that I discussed in the last two chapters were centered on the problem of identifying the source of an infection over a network which is assumed to be know perfectly in advance. On the other hand, the problem of reconstructing propagation networks from observations is a paramount fundamental inverse problem, which is crucial to understand and control the dynamics in complex systems. The methods proposed so far in the literature heavily rely on the complete knowledge of the dynamical trajectories of some spreading process. In certain cases, when information about the time-series of the process is available, the problem can be, and has been, cast into relatively simple terms, as a sequence of time-consecutive states of a pair of nodes gives direct information about the potential interaction between them. In many cases, however, the set of available observations is much sparser, possibly on a much slower timescale than that of the dynamics and often skipping the initial stages of the propagation which would give precious information about the source. In particular, in a single snapshot of the system there is no direct information about the interaction of nodes, as evidence of interaction indeed comes from variation of the state of nodes in time.

Let us take the example of second messenger cascades in a cell, and suppose the experimenter has access to the expression profile of a huge number of proteins in different cascades. Monitoring the exact time course of the concentration of each protein is currently challenging, if not infeasible: one observes a concerted up and down-regulation of a big number of proteins, which naturally follow from a complex time course in a network of reciprocal protein-protein interactions. One is faced with a similar information shortage in the context of epidemic spreading: the plague spreader is unknown, and little is known about the underlying networks of contacts between individuals, which may even be dynamically changing over time.

Reconstructing the connectivity structure of the network is then unavoidably coupled to that of tracing back in time the entire time course of the spreading process so as to locate the source of diffusion. In this chapter, I will give a short account of a work I've been doing in collaboration with A. Braunstein, which led to the development of a Belief Propagation technique to accurately model the posterior distribution of trajectories of the propagation process, and that allows to compute, and then maximize, the likelihood of a given network structure and propagation probabilities [**93**]. We called this method Inverse Dynamics Network Reconstruction (IDNR). Fig. 8.0.1 shows a cartoon representation of the problem. The method is shown to work successfully on several synthetic and real networks, inferring simultaneously the networks and the sources of infection based on sparse observations, including single snapshots, even partial or noisy.

8.1. Method

Let us consider a weighted undirected graph $G = (V, E, \Lambda)$ where $\Lambda = \{\lambda_{ij}\}_{ij \in E}$ play the role of edge-dependent infection probabilities in a SIR stochastic model defined on the network, which is also

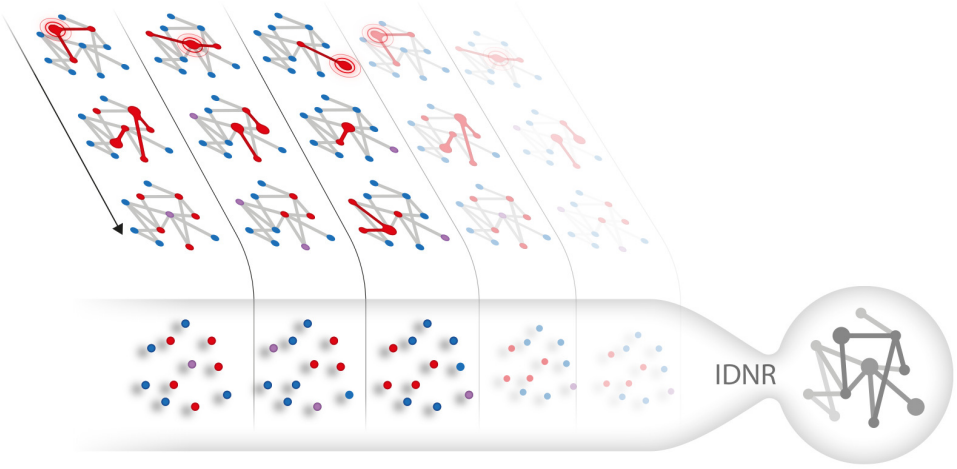

FIGURE 8.0.1. Cartoon representation of the Inverse Dynamics Network Reconstruction method. Each parallel strip stand for an independent cascade in an unknown network, starting from a different source. For every cascade, only a single snapshot is available. The goal is to reconstruct the structure of the network together with the infection probabilities on each edge, and to discover the sources of each cascade as well.

equipped with a set of site-dependent recovery probabilities μ_i. Let us then suppose that a number M of independent realizations (or cascades) of the SIR dynamics are given but only a limited information is available for each of them. Using the same terminology of chapter 6, each cascade is defined by the set of vectors $\mathbf{x_n^t}$, where $n = 1...M$ labels the cascade, and we assume that for each cascade the initial state $\mathbf{x_n^0}$ is composed of just one infected node i_0^n, with all the other node in the network being in the Susceptible state. Let us then assume that we have access to the state of the nodes in the networks only in $T^n = T$ steps after the initiation of each cascades. It is not necessary to constraint a fixed observation delay $T^n = T$, and T^n could be inferred as well for each cascade.

Let us then construct a graphical model representation of the process for each cascade, following the lines of section 6.1, and write down the full posterior probability distributions over the set of cascades. Introducing a weight $\zeta_i^{n,T}$ for each of the n observations, the full distribution is just the product over all the single probability weights 6.1.7 for each cascade, because of the independence

assumption, and takes the form

$$\mathcal{P}\left(\left\{\mathbf{x}_n^0\right\} \mid \left\{\mathbf{x}_n^T\right\}\right) \propto \prod_{n=1}^{M} \sum_{\mathbf{t_n},\mathbf{g_n}} \mathcal{P}\left(\mathbf{x_n}^T | \mathbf{t_n}, \mathbf{g_n}\right) \mathcal{P}\left(\mathbf{t_n}, \mathbf{g_n} | \mathbf{x_n}^0\right) \mathcal{P}\left(\mathbf{x}_n^0\right) \tag{8.1.1}$$

$$= \prod_{n=1}^{M} \sum_{\mathbf{t_n},\mathbf{g_n},\mathbf{s}_n} \prod_{ij} \omega_{ij}^n \prod_i \phi_i^n \mathcal{G}_i^n \gamma_i^n \zeta_i^{n,T} \zeta_i^{n,0} \tag{8.1.2}$$

where all the factors have been labeled with an extra cascade index n.

There is, however, a remarkable difference with equation 6.1.7: since we have no a priori information on the graph topology, the product in the term $\prod_{ij} \omega_{ij}^n$ runs over all the possibile pair i and j in the set V, so that we always work in the setting of a fully connected network with weights λ_{ij}. The main idea of our new approach is that, if the number of cascades M is large enough to convey a reasonable amount of information, the non zero elements of the matrix λ_{ij} will signal the true links in the original graph; in addition, their value will be informative of the heterogeneity of infection probabilities.

Up to now, we are faced with the two-fold problem of inferring the initial source of the spreading for each cascade n together with the unknown structure of the graph. The strategy is the same as in section 6.4: once the log-likelihood $L(\lambda, \mu) = -f(\lambda, \mu) = \log \mathcal{P}\left(\left\{\mathbf{x}_n^T\right\} \mid \{\lambda_{ij}\}, \{\mu_i\}\right)$ with respect to the network parameters is defined, we can use the Bethe decomposition 6.4.2 and equations 6.4.3, 6.4.4 and 6.4.5 to efficiently compute an estimate which only depends upon cavity messages. We then couple the BP updates for the distribution 8.1.1 to a Gradient Ascent (GA) procedure with respect to each network parameter, with the simple equations

$$\lambda_{ij} \leftarrow \lambda_{ij} + \epsilon \frac{\partial L}{\partial \lambda_{ij}} \tag{8.1.3}$$

$$\mu_i \leftarrow \mu_i + \epsilon \frac{\partial L}{\partial \mu_i} \tag{8.1.4}$$

The resulting BP and GA equations take exactly the same form as in Appendix B.1 and B.2, respectively. Once again, we found that it is not necessary to wait for convergence of the BP equations, but a simple alternation suffices to provide good joint estimates for the patient zero in each cascade together with a remarkably good reconstruction of the underlying network.

We can clearly use all the inference machinery that was introduced in section 6.3 to account for noisy, incomplete or confused observations. In addition, the parametrization of the dynamical trajectories is so general that it is possible to implement, with the use of simple evidence terms $\zeta_i^{n,t}$, any kind of observation paradigm: for instance, we can easily deal with a setting in which observations are not constrained to be at a fixed time T but are sparsely distributed in a certain time period (t_0, t_1), so that at each time step a small fraction of nodes in the network is observed.

As a final comment, I also note that, as in the previous sections, the strict condition of a single infected source may be slightly relaxed with an adjustment of the factor γ in the prior $\mathcal{P}\left(\mathbf{x}^0\right) = \prod_n \prod_i \gamma_i\left(x_{n,i}^0\right)$, so as to consider cases in which a small number of nodes are infected at time $t = 0$ in each cascades. Moreover, the use of a fixed duration T for each cascade is not necessary, as one can simply consider a small non zero probability of self infection, and set a maximum time range $(0, T_{max})$, letting the algorithm itself discover the time of self-infection of the source in any given cascade.

Keeping all these straightforward generalizations in mind, in what follows I will focus on the simple setting of M independent snapshots of the entire network, each one being taken after T steps from the initiation in each one of the M observed cascades.

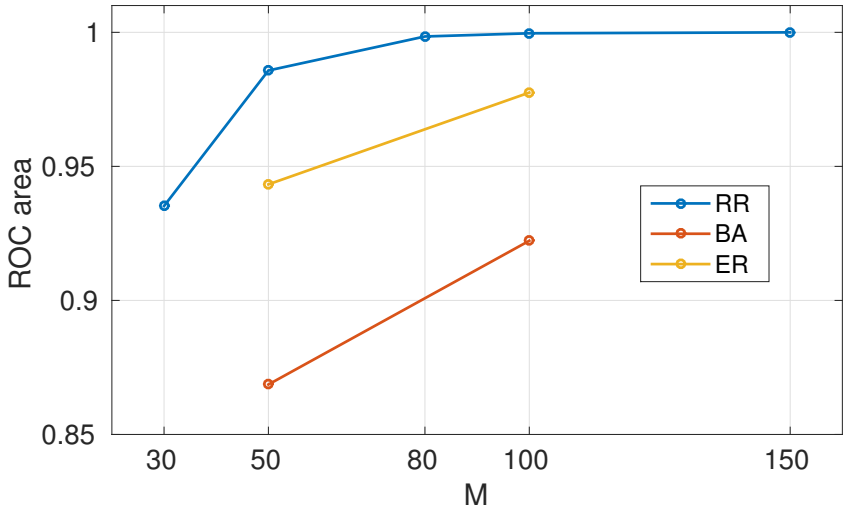

FIGURE 8.2.1. Area under the ROC curve for three different ensembles of random graphs at an increasing number of cascades M. Each point is the average ROC area over 30 instance of the random graph in a given ensemble. Epidemics are run for $T = 5$ and with $\lambda = 0.6$, $\mu = 0.4$. Blue: Random Regular graph with $N = 50$ nodes and $E = 100$ edges. Yellow: Barabasi-Albert graph with $N = 50$, $E \sim 100$. Red: Erdos-Renyi graph with $N = 50$, $E = 100$.

8.2. Results

I start by investigating three basic random network structures, namely Random Regular (RR), Erdos-Renyi (EI) and scale-free networks: an impressive level of accuracy may be reached with a small number M of observations. As a first step, a random graph is constructed, and a set of M cascades are simulated, each one being an independent realization of the stochastic SIR process with a random initial source i_0^n. The IDNR algorithm is then run until the parameters λ_{ij} and μ_i reach a stable value. Since the goal of the inference is two-fold, I use two different measures of the inference performance.

To evaluate the reconstruction performance, I again make use of the *Receiver Operating Characteristic* (*ROC*) curve: the values of λ_{ij} are ranked, and one step upward in the *ROC* is taken if the link is present in the original graph or one step rightward if the link is non existent. Once again, a *ROC* area close to 1 signals a good discrimination between true and non existent links, and it is reported in Fig. 8.2.1. As an examples of the discrimination ability, I report in Fig. 8.2.2 (top) the distribution of inferred weights λ_{ij} for links that exist in the original graphs versus non-existent links: the two are clearly very well separated, confirming the results from the area under the ROC curve. The ability to identify the sources of the spreading is easily quantified by the rank of the true patient zero i_0^n in each of the M cascade: an example is shown in Fig. 8.2.2 (bottom), where it is evident that IDNR successfully recovers most of the true patient zero in each cascade. I also tested the IDNR algorithm on several instances of real interaction networks. As an illustrative example, I performed some experiments on the famous Zachary's Karate Club network, a small social network consisting of $|G| = 34$ nodes and $|E| = 78$ edges, documenting the pairwise interactions over the course

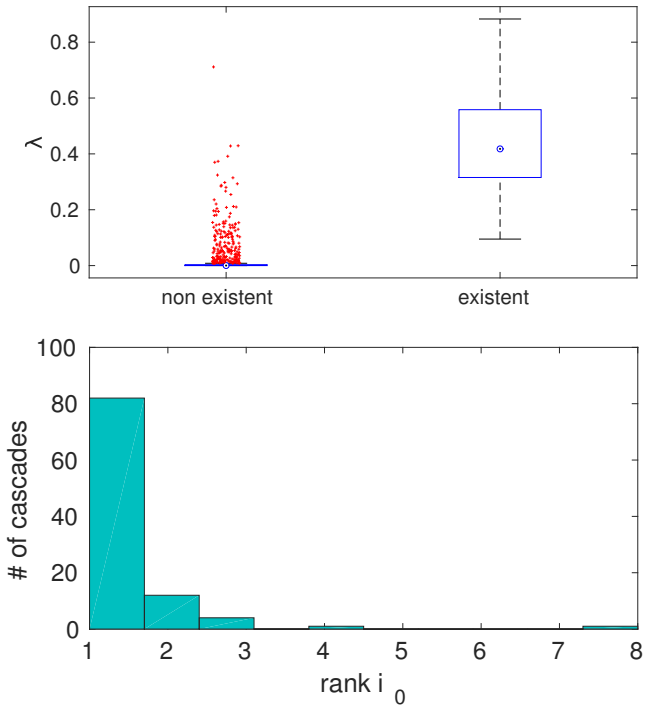

FIGURE 8.2.2. Top: boxplot representation of the inferred λ_{ij} weights for true links that exist in the original graph (right) versus non existent links. Observations come for a set of $M = 100$ cascades in a Random Regular graph with $N = 50$ nodes and $E = 100$ links. Epidemic parameters are $\lambda_{ij} = \lambda = 0.6$ and $\mu_i = \mu = 0.4$. Bottom: histogram of the ranks of the true patient zero i_0^n in each of the $M = 100$ cascades.

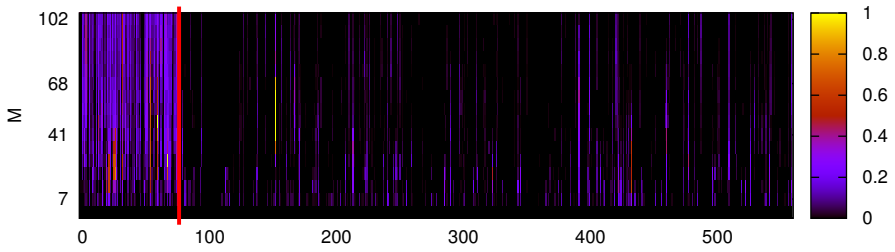

FIGURE 8.2.3. Color plot of inferred values of λ_{ij} in the Zachary's Karate Club network with an increasing number M of cascades (y axis), with epidemic parameters$\lambda = 0.3$ and $\mu = 0.4$. Edges are arranged on the x axis so that the first 78 are on the far left, separated from non existent edges by the read vertical line.

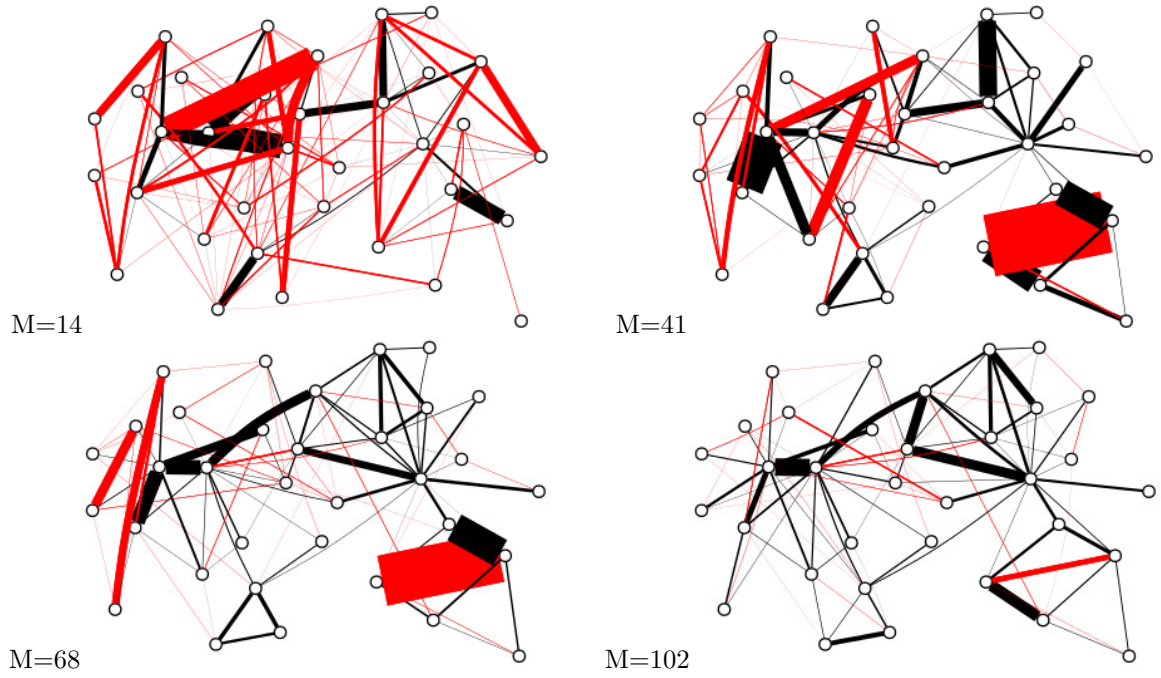

FIGURE 8.2.4. Pictorial representation of the IDNR inference performance in Zachary's Karate Club network with an increasing number M of cascades, with epidemic parameters$\lambda = 0.3$ and $\mu = 0.4$. An edge is thrown between node i and node j if λ_{ij} in non zero, the width of the edge being proportional to the valued λ_{ij}. The true links are colored in black, red links are not present in the original network.

of three years between members of a university-based karate club [**94**]. In this case, I simulated up to $M = 102$ cascades and investigated the performance of the inference method with homogeneous parameters $\lambda = 0.3$ and $\mu = 0.4$ at increasing M.

The results are shown in Fig. 8.2.3: non existent links are clearly distinguished from the true ones even for very small values of M. Fig. 8.2.4 shows a pictorial representation of the inference method as the number of cascades increases, the black links being the ones actually present in the original graph, whereas the red color signals non existent links (the links width is proportional to the inferred value λ_{ij}). The method was tested on larger networks with similar results. Results will be presented in an upcoming publications [**93**].

8.3. Summary

In this chapter, I gave a very brief account of a recent development of the Bayesian approach which was introduced in the last two chapters, which is capable of solving a complex inverse problem and to extract interesting functional and topological information, under a very limited amount of static observations. The method is extremely effective in reconstructing the topology of the hidden network even when the number of snaphots is very small. It can be used for inference from several independent cascades with partial observations than can be also noisy. There are several advantages of this approach over existing ones. The main one is that several inference problems can be treated

under a unique formulation. I will speculate more on the possible extensions and applications of the method in chapter 9.

CHAPTER 9

Conclusions

In this Thesis, a number of techniques from the Statistical Physics of Spin Glasses have been used to deal with complicated inverse problems. The main focus has been on inference and learning problems defined on complex network: all in all, they fall in the vast realm of inverse problems, that can be tackled with methods able to deal with the analysis of complicated probability distributions characterized by some kind of local factorization structure.

As I discussed extensively, analytical and computational methods borrowed from the Statistical Mechanics of Disordered Systems can be profitably used non only to study analytically the typical case, but also, more importantly, to construct solving algorithms that can tackle single instances. The case of the perceptron learning in chapter 4 is a very good example of the power of the concurrent use of Replica theory in the typical case and Belief Propagation (with a set of heuristic generalizations) to solve a hard constraint satisfaction problem and to characterize the properties of the provided solutions.

Subdominant states and EdMC. In the first part of the Thesis, the relevance of subdominant states in the perceptron learning problem has been demonstrated, both in the classification and the generalization case. A series of algorithmic results have been shown to be in contrast with the standard picture of a glassy landscape of isolated solutions. A new method for analyzing the local structure of the space of solutions has been presented: the analytical technique is built on an entropic re-weighting term in the standard Replica calculation of the free energy. In chapter 4 it has been shown that the properties of a number of algorithmic solvers can be understood in terms of a subdominant set of solutions which are embedded in a large cluster, very dense at its core. The features of solutions inside the cluster are peculiar, in that they show a remarkably smaller generalization error with respect to the typical ones in the teacher-student scenario.

There are a number of important points that are currently addressed as interesting generalization of the approach presented here:

- the generalization of the analytical calculation carried in the binary case to a more realistic setting where synaptic weights can take a finite number of discrete states, as well as the case where the patterns are sparse — preliminary results show that the qualitative picture is almost identical [**58**];
- the search for on-line learning methods which can be proved to be fully described by the out-of-equilibrium measure of section 4.2;
- finally, an investigation of the relevance of the subdominant cluster in the context of attractor neural networks, which serve as simple neural models for associative memory.

In chapter 5 the ideas raised in the investigation of the perceptron learning problem have been used to introduce a novel MCMC method which, at odds with standard Monte Carlo methods, is not guided by the energy function, but uses the local entropy as a measure of local density of solutions. It has been shown that local entropy provides a smoother landscape and the correspondent Entropy driven Monte

Carlo (EdMC) has been shown to be immune from trapping in local minima, and have nice scaling properties. In the particular case of the perceptron, the properties of solutions that EdMC is able to find have been analyzed by means of a Replica calculation where, analogously to the algorithmic procedure, no constraint on the reference configuration is imposed. The analytical results match very well the behavior of the algorithm, then proving the capability of a simple entropy driven MCMC approach to effectively guide the system in a state which is surrounded by a large number of zero energy configurations. EdMC has been tested in a completely different dilute setting, the random K-satisfiability problem, yielding good results even in the regions of parameters which are known to be hard for standard algorithm.

At present, EdMC relies on a Bethe approximation for the local entropy, which means that for each proposed move the Metropolis algorithm has to wait for Belief Propagation to converge, with a set of external fields corresponding to the current state of the system. This could eventually be a limiting factor when the method has to be scaled to problems with a very large number of variables. In particular, the most interesting problem is how to devise efficient methods for inferring the local entropy without resorting to a BP approach. This is the subject of current research.

Inverse dynamics in epidemics and network reconstruction. In chapter 6 a previously introduced Bayesian method for inferring the patient zero in an SIR model of epidemic spreading over networks has been generalized to accont for uncertainty in observations and epidemic parameters as well. It has been shown that a good inference of the hidden epidemic parameters can be achieved if one couples the Belief Propagation updates with a simple on-line likelihood maximization method.

In chapter 7 the inference machinery has been shown to be generalizable to the case of a continuous-time model of SIR spreading, with a technique that is now capable to deal with real time measurements of contacts inside a community.

In chapter 8 a very recent and promising generalization of the inference approach of chapter 6 has been presented. The method, called Inverse Dynamics Network Reconstruction (IDNR), is an inference technique which couples the problem of identifying the source of the spreading in a series of cascades to that of reconstructing the topological structure of a network. The inference method relies solely on a single observation for each cascade, and its performance is striking even when the algorithm is supplied with a very limited information on the dynamics. Owing to the generality of the Bayesian method, the technique is applicable to a wide variety of irreversible spreading processes with a stochastic dynamics. The method is effective on progressive propagation models like susceptible-infected (SI), susceptible-infected-removed (SIR), independent cascades (IC) and variants, including models with hidden variables (e.g. representing latency times). Moreover, there are a number of interesting generalizations: the simplest extension would be the (random) Bootstrap Percolation case where each node gets activated only when it receives inputs from a given (in general site-dependent) number of neighbors. These models are used to describe the features of dynamical processes in neuronal networks. Work is in progress in these directions.

APPENDIX A

EdMC: analytical details

In this section, I will present a different form of the BP equations, which will be written in a different parametrization for the messages, I will define the quantities used by EdMC, and explicitly write the relevant expressions for the cases of the Binary Perceptron and the K-SAT problems. I will make use of the notation introduced in Sec. 5.1.

A.1. General BP scheme

The modified system defined by the Hamiltonian $H(x;\tilde{x})$ of eq. (5.1.4) introduces an additional external field term in the BP equations (3.1.2,3.1.3), that read:

$$\nu_{i\to\mu}(x_i;\tilde{x}_i) \propto e^{\gamma\tilde{x}_i x_i}\prod_{\nu\in\partial i\setminus\mu}\hat{\nu}_{\nu\to i}(x_i) \tag{A.1.1}$$

$$\nu_{i}(x_i;\tilde{x}_i) \propto e^{\gamma\tilde{x}_i x_i}\prod_{\nu\in\partial i}\hat{\nu}_{\nu\to i}(x_i) \tag{A.1.2}$$

It will be helpful to give also the expression for the cavity magnetization fields in absence of the external fields (in the case of binary spins $x_i=\pm 1$):

$$h_i = \tanh^{-1}\left(\nu_i(1)-\nu_i(-1)\right) \tag{A.1.3}$$

These will be central in the heuristic version of EdMC presented in Sec. 5.2.2 (Algorithm A.3).

The local free entropy $F(\tilde{x},\gamma)$ of eq. (5.1.2) can be computed from the fixed point messages in the zero-temperature limit $\beta\to\infty$ in terms of purely local contributions from variables, edges and factor nodes:

$$F(\tilde{x},\gamma)=\sum_{\mu}\left(F_\mu(\tilde{x},\gamma)+\sum_{i\in\partial\mu}F_{i\to\mu}(\tilde{x},\gamma)\right)-\sum_i\left(|\partial i|-1\right)F_i(\tilde{x},\gamma) \tag{A.1.4}$$

where:

$$F_\mu(\tilde{x},\gamma) = \log\left(\sum_{\{x_{\partial\mu}:E(x_{\partial\mu})=0\}}\prod_{i\in\partial\mu}\nu_{i\to\mu}(x_i;\tilde{x}_i)\right) \tag{A.1.5}$$

$$F_{i\to\mu}(\tilde{x},\gamma) = \log\left(\sum_{x_i}e^{\gamma\tilde{x}_i x_i}\prod_{\nu\in\partial i\setminus\mu}\hat{\nu}_{\nu\to i}(x_i)\right) \tag{A.1.6}$$

$$F_{i}(\tilde{x},\gamma) = \log\left(\sum_{x_i}e^{\gamma\tilde{x}_i x_i}\prod_{\mu\in\partial i}\hat{\nu}_{\mu\to i}(x_i)\right) \tag{A.1.7}$$

The overlap $S(\tilde{x},\gamma) = \frac{1}{N}\langle \tilde{x}\cdot x\rangle$ and the local entropy $\mathscr{S}(\tilde{x},\gamma)$ can be computed as:

$$S(\tilde{x},\gamma) = \frac{1}{N}\sum_i \tilde{x}_i \sum_{x_i} x_i \nu_i(x_i;\tilde{x}_i) \tag{A.1.8}$$

$$\mathscr{S}(\tilde{x},\gamma) = F(\tilde{x},\gamma) - \gamma S(\tilde{x},\gamma) \tag{A.1.9}$$

These expressions are used in Sec. 5.2.3, where $F(\tilde{x},\gamma)$ is optimized over $\tilde{x}$ and they are averaged over many realizations of the patterns to compare them to the theoretical expression of eq. (5.1.6).

A.1.1. BP for the binary perceptron. The BP equations for a given instance (ξ^μ,σ^μ), $\mu = 1...\alpha N$ is most easily written in terms of cavity magnetizations $m_{i\to\mu} \propto \nu_{i\to\mu}(+1) - u_{i\to\mu}(-1)$ and $\hat{m}_{\mu\to i} \propto \hat{\nu}_{\mu\to i}(+1) - \hat{\nu}_{\mu\to i}(-1)$. It is always possible to set $\forall\mu : \sigma^\mu = 1$ without loss of generality, by means of the simple gauge transformation $\xi_i^\mu \to \sigma^\mu \xi_i^\mu$. With these simplification, equations (3.1.2,3.1.3) become:

$$m_{i\to\mu} = \tanh\left(\sum_{\nu\neq\mu} \tanh^{-1}(\hat{m}_{\nu\to i})\right) \tag{A.1.10}$$

$$\hat{m}_{\mu\to i} = \frac{\sum_{s=-\xi_i}^{N-1} D_{\mu\to i}(s) - \sum_{s=\xi_i}^{N-1} D_{\mu\to i}(s)}{\sum_{s=-\xi_i}^{N-1} D_{\mu\to i}(s) + \sum_{s=\xi_i}^{N-1} D_{\mu\to i}(s)} \tag{A.1.11}$$

where

$$D_{\mu\to i}(s) = \sum_{\{x_j\}_{j\neq i}} \delta\left(s, \sum_j x_j\xi_j\right) \prod_{j\neq i} \frac{(1+x_j m_{j\to\mu})}{2} \tag{A.1.12}$$

is the convolution of the all cavity messages $m_{j\to\mu}$ impinging on the pattern node μ, except for $m_{i\to\mu}$: it is thus the (cavity) distribution of the total synaptic input for pattern μ, in absence of the synapse i. The complexity of the second update is at most $O\left(N^2\right)$ with an appropriate pre-computation of cavity convolutions. As I pointed out in section 4.1.1, when one deals with the case of $N \gg 1$ and an extensive number of patterns, a common and simple strategy is to adopt a Gaussian approximation $\tilde{D}_{\mu\to i}(s) = \frac{1}{b_{\mu\to i}} G\left(\frac{s-a_{\mu\to i}}{b_{\mu\to i}}\right)$ for the distribution $D_{\mu\to i}(s)$, where $G(s)$ denotes the normal distribution. It suffices then to compute the mean $a_{\mu\to i}$ and variance $b^2_{\mu\to i}$ of the distribution $\tilde{D}_{\mu\to i}(s)$, whose dependence on cavity messages $m_{j\to\mu}$ is easily determined from the central limit theorem:

$$a_{\mu\to i} = \sum_{j\neq i} \xi_j^\mu m_{j\to\mu} \tag{A.1.13}$$

$$b^2_{\mu\to i} = \sum_{j\neq i} \left(1 - m^2_{j\to\mu}\right) \tag{A.1.14}$$

(analogous non-cavity quantities a_μ and b_μ are computed by summing over all indices j). By doing so, equation (A.1.11) becomes:

$$\hat{m}_{\mu\to i} = \xi_i\, g(a_{\mu\to i}, b_{\mu\to i}) \tag{A.1.15}$$

where

$$g(a,b) = \frac{H\left(\frac{a-1}{b}\right) - H\left(\frac{a+1}{b}\right)}{H\left(\frac{a-1}{b}\right) + H\left(\frac{a+1}{b}\right)} \tag{A.1.16}$$

and we used the function $H(x) = \frac{1}{2}\text{erfc}\left(\frac{x}{\sqrt{2}}\right)$.

The free-entropy F_{perc} can be easily obtained by eq. (A.1.4) putting $\gamma = 0$. In the Gaussian approximation, the expression can be written as:

$$\begin{aligned} F_{\text{perc}} &= \sum_{\mu} \log\left(H\left(\frac{a_\mu}{b_\mu}\right)\right) - \sum_{i,\mu} \log\left(1 + m_{i\to\mu}\hat{m}_{\mu\to i}\right) \\ &\quad + \sum_{i} \log\left[\prod_{\mu}\left(1+\hat{m}_{\mu\to i}\right) + \prod_{\mu}\left(1-\hat{m}_{\mu\to i}\right)\right] \end{aligned} \tag{A.1.17}$$

The total number of solutions for a given instance can be determined by means of the entropy:

$$\begin{aligned} \mathscr{S}_{\text{perc}} &= \sum_{\mu} \log\left(H\left(\frac{a_\mu}{b_\mu}\right)\right) - \sum_{i,\mu}\left[\frac{1+m_i}{2}\log\left(\frac{1+m_{i\to\mu}}{2}\right) + \frac{1-m_i}{2}\log\left(\frac{1-m_{i\to\mu}}{2}\right)\right] \\ &\quad + (M-1)\left[\frac{1+m_i}{2}\log\left(\frac{1+m_i}{2}\right) + \frac{1-m_i}{2}\log\left(\frac{1-m_i}{2}\right)\right] \end{aligned} \tag{A.1.18}$$

Consistently with the theoretical predictions, BP equations always converge for $\alpha < \alpha_c$. Indeed, Replica Symmetry holds with the identification of states as the dominant isolated configurations, this being evident in the proportionality relation between RS and RSB free-energy [**24**], with an intra-state overlap (RSB parameter) q_1 equal to 1. The entropy decreases monotonically with α and, provided N is high enough, it vanishes at the critical threshold $\alpha_c \sim 0.833$ [**24**].

As I said in section 4.1.2, if one slightly modifies the original equations with the introduction of a 'reinforcement term', much similar to a sort of smooth decimation procedure, BP becomes a very efficient solver that is able to find a solution with probability 1 up to $\alpha \sim 0.74$ [**52**]. Further simplifications of these equations lead to the SBPI [**53**] and CP+R [**54**] algorithms.

Introducing the external fields $\gamma\tilde{x}_i$ of eq. (5.1.4) is very simple (cf. eqs. (A.1.1) and (A.1.2)). Eq. (A.1.10) for the cavity magnetization is modified as:

$$m_{i\to\mu} = \tanh\left(\sum_{\nu\neq\mu}\tanh^{-1}\left(m_{\nu\to i}\right) + \gamma\tilde{x}_i\right) \tag{A.1.19}$$

and similarly for the total magnetization: $m_i = \tanh\left(\sum_\mu \tanh^{-1}\left(\hat{m}_{\mu\to i}\right) + \gamma\tilde{x}_i\right)$. The local free entropy is also simply given by the expression $F_{\text{perc}}(\tilde{x},\gamma) = \mathscr{S}_{\text{perc}}(\tilde{x},\gamma) + \gamma S(\tilde{x},\gamma)$ using eqs. (A.1.18) and (A.1.8) with the modified magnetizations. The cavity magnetization fields in absence of the external fields, eq. (A.1.3), are:

$$h_i = \sum_{\mu}\tanh^{-1}\left(m_{\mu\to i}\right) \tag{A.1.20}$$

A.1.2. BP for K-SAT. The Belief Propagation equations for the K-SAT problem are most easily written with a parametrization of the BP messages$\{\nu_{i\to\mu}, \hat{\nu}_{\mu\to i}\}$ in terms of the quantities $\{\zeta_{i\to\mu}, \eta_{\mu\to i}\}$, where $\zeta_{i\to\mu}$ is the probability that x_i does not satisfy clause μ in absence of this clause, and $\eta_{\mu\to i}$ is the probability that all the variables in clause μ except i violate the clause. With this choice, and calling $V(\mu)$ the set of variables in the constraint μ, equation (3.1.3) simply becomes:

$$\eta_{\mu\to i} = \prod_{j\in V(\mu)\setminus i}\zeta_{j\to\mu} \tag{A.1.21}$$

Equation (3.1.2) for the variable node update is slightly more involved:

$$\zeta_{i\to\mu} = \frac{\prod_{\rho\in V_\mu^s(i)}(1-\eta_{\rho\to i})}{\prod_{\rho\in V_\mu^s(i)}(1-\eta_{\rho\to i}) + \prod_{\rho\in V_\mu^u(i)}(1-\eta_{\rho\to i})} \tag{A.1.22}$$

where $V_\mu^s(i)$ (resp. $V_\mu^u(i)$) is the set of clauses ρ in which variable i is involved with a coupling $J_i^\rho \neq J_i^\mu$ (resp. $J_i^\rho = J_i^\mu$).

As for the perceptron, the free-entropy is obtained from expression (A.1.4) at $\gamma = 0$:

$$\begin{aligned} F_{\rm SAT} &= \sum_\mu \log\left(1 - \prod_{i\in V(\mu)}\zeta_{i\to\mu}\right) - \sum_i\sum_{\mu\in\partial i}\log\left(1-\zeta_{i\to\mu}^2\right) + \\ &+ \sum_i \log\left[\prod_{\rho\in V^+(i)}(1-\eta_{\rho\to i}) + \prod_{\rho\in V^-(i)}(1-\eta_{\rho\to i})\right] \end{aligned} \tag{A.1.23}$$

where $V^+(i)$ (resp. $V^-(i)$) is the set of all clauses μ in which variable i is involved with $J_i^\mu = -1$ (resp. $J_i^\mu = 1$). Analogously, the entropy is obtained from the following expression, which only depends upon the messages $\eta_{\mu\to i}$:

$$\begin{aligned} \mathscr{S}_{\rm SAT} &= \sum_\mu \log\left[\prod_{i\in V(\mu)}\left(\prod_{\rho\in V_\mu^s(i)}(1-\eta_{\rho\to i}) + \prod_{\rho\in V_\mu^u(i)}(1-\eta_{\rho\to i})\right) - \prod_{i\in V(\mu)}\left(\prod_{\rho\in V_\mu^u(i)}(1-\eta_{\rho\to i})\right)\right] \\ &+ \sum_i (1-n_i)\log\left[\prod_{\rho\in V^+(i)}(1-\eta_{\rho\to i}) + \prod_{\rho\in V^-(i)}(1-\eta_{\rho\to i})\right] \end{aligned} \tag{A.1.24}$$

In the chosen parametrization, the external fields $\gamma\tilde{x}_i$ may be easily introduced by means of N additional single-variable 'soft clauses' $\tilde{C}_i$, which send fixed cavity messages to their respective variables, taking the value:

$$\eta_{\tilde{C}_i\to i} = \frac{2\tanh\gamma}{1+\tanh\gamma} \tag{A.1.25}$$

with the restriction that $\tilde{C}_i \in V^+(i)$ (resp. $\tilde{C}_i \in V^-(i)$) if $\tilde{x}_i = +1$ (resp. $\tilde{x}_i = -1$).

Calling $\tilde{V}^+(i)$, $\tilde{V}^-(i)$ the new set of clauses, enlarged so as to contain the $\tilde{C}_i$, the total magnetization is given by:

$$m_i = \frac{\prod_{\rho\in\tilde{V}^-(i)}(1-\eta_{\rho\to i}) - \prod_{\rho\in\tilde{V}^+(i)}(1-\eta_{\rho\to i})}{\prod_{\rho\in\tilde{V}^-(i)}(1-\eta_{\rho\to i}) + \prod_{\rho\in\tilde{V}^+(i)}(1-\eta_{\rho\to i})} \tag{A.1.26}$$

The cavity magnetization fields in absence of the external fields, eq. (A.1.3), are simply given by:

$$h_i = \tanh^{-1}(m_i) - \gamma\tilde{x}_i \tag{A.1.27}$$

Convergence properties of equations (A.1.21,A.1.22) on single instances are deeply related to the Replica Symmetry Breaking scenario. The onset of long range correlations in clustered RSB phase prevents BP from converging.

While RSB limits the usefulness of BP (as well as reinforced BP) at high α, ideas from the 1-RSB cavity method, when applied to the single case without the averaging process, have led to the

introduction of a powerful heuristic algorithm, Survey Propagation (SP) [**61, 95**], which is able to solve K-SAT instances in the hard phase, almost up to the UNSAT threshold.

A.2. Details of the out of equilibrium analysis for the binary Perceptron Learning problem

In this section, I will provide all technical details of the analysis of the reweighted free energy function of eq. (5.1.5) of Sec. 5.1 for the case of the Perceptron Learning problem with binary synapses of section. 5.2.1.

A.2.1. Setting the problem and the notation. Patterns are generated drawing the inputs as random i.i.d. variables $\xi_i \in \{-1, +1\}$ with distribution $P(\xi_i) = \frac{1}{2}\delta(\xi_i - 1) + \frac{1}{2}\delta(\xi_i + 1)$. Without loss of generality, the desired output of all patterns is 1. The problem of correctly classifying a set of αN patterns $\{\xi^\mu\}$ (where $\mu = 1, \dots, \alpha N$) is addressed defining, for any vector of synaptic weights $W = \{W_i\}_{i=1,\dots,N}$ with $W_i \in \{-1, +1\}$ the quantity

$$\mathbb{X}_\xi(W) = \prod_\mu \Theta\left(\frac{1}{\sqrt{N}} \sum_i W_i \xi_i^\mu\right) \tag{A.2.1}$$

(with Θ the Heaviside step function: $\Theta(x) = 1$ if $x \geq 0$, 0 otherwise) such that the solutions of the learning problem are described by $\mathbb{X}_\xi(W) = 1$.

Throughout this section, I will use the index i for the synapses and μ for the patterns. In all sums and products, the summation ranges are assumed implicitly, e.g. $\sum_i \equiv \sum_{i=1}^N$. Also, all integrals are assumed to be taken over $\mathbb{R}$ unless otherwise specified. The letters a, b, c and d will be reserved for replica indices (see below). Finally, when introducing the 1-RSB Ansatz, I will also use α, β, α' and β' for the replica indices; it should be clear from the context that these do not refer to the capacity α or the inverse temperature β in those cases.

The number of solutions is given as:

$$\mathcal{N}_\xi = \sum_{\{W\}} \mathbb{X}_\xi(W) \tag{A.2.2}$$

Let us then consider a set of reference configurations, each of which has an associated set of solutions that are constrained to have a certain overlap S with it. The reference configurations will be indicated by the vector $\tilde{W}$, W are all the other solutions.

Let us define then:

$$\mathcal{N}_\xi\left(\tilde{W}, S\right) = \sum_{\{W\}} \mathbb{X}_\xi(W)\, \delta\left(\sum_i W_i \tilde{W}_i - SN\right) \tag{A.2.3}$$

(where δ is the Dirac-delta distribution), i.e. the number of solutions W that have overlap S with (or equivalently, distance $\frac{1-S}{2}$ from) a reference configuration $\tilde{W}$. The goal is to compute the following quenched free energy:

$$\mathscr{F}(S, y) = -\frac{1}{Ny} \langle \log(\Omega(S, y)) \rangle_{\{\xi^\mu\}} = -\frac{1}{Ny} \left\langle \log\left(\sum_{\{\tilde{W}\}} \mathcal{N}_\xi\left(\tilde{W}, S\right)^y\right) \right\rangle_{\{\xi^\mu\}} \tag{A.2.4}$$

where $\Omega(S,y)$ is the partition function and y has the role of an inverse temperature. This is the free energy density of a system where the configuration is described by $\tilde{W}$ and the energy is given by minus the entropy of the other solutions with overlap S from it.

Note that this expression is almost equivalent to eq. 5.1.5, except that the overlap S is used as a control parameter instead of the coupling γ, by using a hard constraint (the delta distribution in eq. A.2.3) rather than a soft constraint (the exponential in eq. 5.1.3). The two parameters are conjugates. The main advantage of using γ is that it is easier to implement using the BP algorithm, which is why it was used for the EdMC algorithm. The advantage of using S is that it provides a more general description: while in large portions of the phase space the relationship between γ and S is bijective (and thus the two systems are equivalent), some regions of the phase space at large α can only be fully explored with by constraining S, and thus we have used this system for the theoretical analysis.

The main goal is that of studying the ground states of the system, i.e. taking the limit of $y \to \infty$. This limit makes possible to seek the reference configuration $\tilde{W}$ for which the number of solutions at overlap S with it is maximal, and to derive an expression for the entropy $\mathscr{S}(S,y) = \left\langle \log \mathcal{N}_\xi \left(\tilde{W}, S\right)\right\rangle$ of the surrounding solutions, which was called local entropy. It will turn out that $\mathscr{S}(S,\infty)$ is always positive for $\alpha < \alpha_c$ when $S \to 1$, indicating the presence of dense clusters of solutions.

In the remainder, I will generally use the customary notation $\int d\mu(W)$ or $\int \prod_i d\mu(W_i)$ (instead of $\sum_W$ or $\sum_{\{W\}}$) to denote the integral over possible values of the weights; for binary weights, one simply has:

$$d\mu(W) = (\delta(W-1) + \delta(W+1))\, dW$$

A.2.2. Entropy and Complexity.

A.2.2.1. *Replica trick.* In order to compute the quantity of eq. (A.2.4), the replica trick will be introduced. In what follows, the letter n will denote the number of replicas of the reference configurations, and the letters c and d, with $c,d \in \{1,\dots,n\}$, will be their replica indices.

Let us then write $\mathcal{N}\left(\tilde{W},S\right)^y$ as a product of y "local" replicas. There will be a different set of local replicas for each replicated reference configuration, so that the local replicas are yn in total. The indices a and b will be used to denote these local replicas (each of which will also have a reference replica index c), i.e. $a,b \in \{1,\dots,y\}$.

Therefore, one needs to compute:

$$\begin{aligned}(\text{A.2.5}) \qquad & \lim_{n\to 0} \langle \Omega(S,y)^n \rangle_{\{\xi^\mu\}} = \\ & = \lim_{n\to 0} \left\langle \int \prod_{ic} d\mu\left(\tilde{W}_i^c\right) \int \prod_{ica} d\mu\left(W_i^{ca}\right) \prod_{ca} \mathbb{X}_\xi\left(W^{ca}\right) \prod_{ca} \delta\left(\sum_i W_i^{ca}\tilde{W}_i^c - SN\right)\right\rangle_{\{\xi^\mu\}}\end{aligned}$$

As a first step, let us substitute the arguments of the theta functions in the $\mathbb{X}_\xi$ terms via Dirac-delta functions:

$$(\text{A.2.6}) \qquad \prod_{ca\mu} \Theta\left(\frac{1}{\sqrt{N}}\sum_i W_i^{ca}\xi_i^\mu\right) = \int \prod_{ca\mu} d\lambda_\mu^{ca}\, \delta\left(\lambda_\mu^{ca} - \frac{1}{\sqrt{N}}\sum_i W_i^{ca}\xi_i^\mu\right) \prod_{ca\mu} \Theta\left(\lambda_\mu^{ca}\right)$$

Then, the delta functions are expanded using their integral representation:

$$(\text{A.2.7}) \qquad \delta\left(\lambda_\mu^{ca} - \frac{1}{\sqrt{N}}\sum_i W_i^{ca}\xi_i^\mu\right) = \int \frac{d\hat{\lambda}_\mu^{ca}}{2\pi} \exp\left(i\hat{\lambda}_\mu^{ca}\lambda_\mu^{ca} - i\hat{\lambda}_\mu^{ca}\frac{1}{\sqrt{N}}\sum_i W_i^{ca}\xi_i^\mu\right)$$

With these, the expression where the patterns are involved can be factorized, so as to compute the averages over the patterns independently for each μ and for each i, and expand for large N:

$$\begin{aligned}
&\prod_i \int \left(P\left(\xi_i^\mu\right) d\xi_i^\mu\right) \exp\left(-\frac{i}{\sqrt{N}}\xi_i^\mu\left(\sum_{ca} W_i^{ca}\hat{\lambda}_\mu^{ca}\right)\right) = \\
&\simeq \exp\left(-\frac{1}{2N}\sum_i\left(\sum_{ca} W_i^{ca}\hat{\lambda}_\mu^{ca}\right)^2\right) \\
&= \exp\left(-\frac{1}{2}\left(\sum_{cadb}\hat{\lambda}_\mu^{ca}\hat{\lambda}_\mu^{db}\left(\frac{1}{N}\sum_i W_i^{ca}W_i^{db}\right)\right)\right) \qquad \text{(A.2.8)}
\end{aligned}$$

Next, the order parameters for the overlaps are introduced via delta functions (the one for the overlaps $\frac{1}{N}\sum W_i^{ca}\tilde{W}_i^c$, which must be equal to S, is already apparent in the expression). Using the expressions (A.2.6), (A.2.7) and (A.2.8) in eq. (A.2.5), one gets:

$$\begin{aligned}
\left\langle \Omega\left(S,y\right)^n\right\rangle_{\{\xi^\mu\}} &= \int \prod_{ic} d\mu\left(\tilde{W}_i^c\right)\int\prod_{ica} d\mu\left(W_i^{ca}\right)\int\prod_{ca\mu}\frac{d\lambda_\mu^{ca}d\hat{\lambda}_\mu^{ca}}{2\pi}\prod_\mu e^{i\left(\sum_{ca}\lambda_\mu^{ca}\hat{\lambda}_\mu^{ca}\right)}\times \\
&\times\int\prod_{c,a>b}\left(dq^{ca,cb}N\right)\delta\left(Nq^{ca,cb}-\sum_i W_i^{ca}W_i^{cb}\right)\times \\
&\times\int\prod_{c>d,ab}\left(dq^{ca,db}N\right)\delta\left(Nq^{ca,db}-\sum_i W_i^{ca}W_i^{db}\right)\times \\
&\times\prod_{ca}\delta\left(NS-\sum_i W_i^{ca}\tilde{W}_i^c\right)\prod_{ca\mu}\Theta\left(\lambda_\mu^{ca}\right)\prod_\mu\exp\left(-\frac{1}{2}\sum_{ca}\left(\hat{\lambda}_\mu^{ca}\right)^2\right)\times \\
&\times\prod_\mu\exp\left(-\sum_c\sum_{a>b}\hat{\lambda}_\mu^{ca}\hat{\lambda}_\mu^{cb}q^{ca,cb}-\sum_{c>d}\sum_{ab}\hat{\lambda}_\mu^{ca}\hat{\lambda}_\mu^{db}q^{ca,db}\right) \qquad \text{(A.2.9)}
\end{aligned}$$

The delta functions are again expanded in the usual way introducing conjugate parameters $\hat{q}^{ca,db}$ and $\hat{S}^{ca}$, and the integrals are rearranged such that the structure is factorizable over μ and over i . One then obtains:

$$\begin{aligned}
\left\langle \Omega\left(S,y\right)^n\right\rangle_{\{\xi^\mu\}} &= \int\prod_{c,a>b}\left(\frac{dq^{ca,cb}d\hat{q}^{ca,cb}N}{2\pi}\right)\int\prod_{c>d,ab}\left(\frac{dq^{ca,db}d\hat{q}^{ca,db}N}{2\pi}\right)\int\prod_{ca}\left(\frac{d\hat{S}^{ca}N}{2\pi}\right) \\
&e^{-N\left(\sum_c\sum_{a>b}q^{ca,cb}\hat{q}^{ca,cb}+\sum_{c>d}\sum_{ab}q^{ca,db}\hat{q}^{ca,db}+\sum_{ca}S\hat{S}^{ca}\right)}G_S^N\,G_E^{\alpha N} \qquad \text{(A.2.10)}
\end{aligned}$$

where G_S and G_E are the entropic and the energetic terms, respectively:

$$\begin{aligned}
G_S &= \int\prod_c d\mu\left(\tilde{W}^c\right)\int\prod_{ca}d\mu\left(W^{ca}\right)\exp\left(\sum_c\sum_{a>b}\hat{q}^{ca,cb}W^{ca}W^{cb}+\right. \qquad \text{(A.2.11)} \\
&\left.+\sum_{c>d}\sum_{ab}\hat{q}^{ca,db}W^{ca}W^{db}+\sum_{ca}\hat{S}^{ca}W^{ca}\tilde{W}^c\right)
\end{aligned}$$

$$G_E = \int \prod_{ca} \frac{d\lambda^{ca} d\hat{\lambda}^{ca}}{2\pi} e^{i\left(\sum_{ca} \lambda^{ca}\hat{\lambda}^{ca}\right)} \prod_{ca} \Theta\left(\lambda^{ca}\right) \exp\left(-\frac{1}{2}\sum_{ca}\left(\hat{\lambda}^{ca}\right)^2 + \right.$$
$$\left. - \sum_{c}\sum_{a>b} \hat{\lambda}^{ca}\hat{\lambda}^{cb} q^{ca,cb} - \sum_{c>d}\sum_{ab} \hat{\lambda}^{ca}\hat{\lambda}^{db} q^{ca,db}\right) \tag{A.2.12}$$

A.2.2.2. *The external* 1*-RSB Ansatz.* As explained in section 5.2.1, a 1-RSB Ansatz for the planted configurations is introduced. More specifically, the n replicas will be divided in $\frac{n}{m}$ groups of m replicas each, with m the Parisi 1-RSB parameter to subsequently optimize upon. Let us then introduce the multi-index $c = (\alpha, \beta)$, where $\alpha \in \{1, ..., n/m\}$ labels a block of m replicas, and $\beta \in \{1, ..., m\}$ is the index of replicas inside the block. This induces the following structure for the overlap matrix $q^{\alpha\beta,a;\alpha'\beta',b} \equiv q^{ca,db}$:

$$q^{\alpha\beta,a;\alpha'\beta',b} = \begin{cases} 1 & \text{if}\, \alpha = \alpha', \beta = \beta', a = b \\ q_2 & \text{if}\, \alpha = \alpha', \beta = \beta', a \neq b \\ q_1 & \text{if}\, \alpha = \alpha', \beta \neq \beta' \\ q_0 & \text{if}\, \alpha \neq \alpha' \end{cases} \tag{A.2.13}$$

The structure of the conjugated parameters matrix $\hat{q}^{ca,db}$ is analogous. I will also assume $\hat{S}^{ca} = \hat{S}$. Note that $\hat{S}$, being the conjugate of S, takes the role of the soft constraint parameter γ of eq. 5.1.3 that is used throughout the main text. In fact, as already noted, studying the soft-constrained system of eq. 5.1.5 gives exactly the same results with $\hat{S} = \gamma$, provided one makes an equivalent symmetric Ansatz on the overlap S as it is done for $\hat{S}$ here.

A.2.2.3. *Entropic term.* Let us consider the entropic term in the 1-RSB Ansatz:

$$G_S = \int \prod_{\alpha\beta} d\mu\left(\tilde{W}^{\alpha\beta}\right) \int \prod_{\alpha\beta,a} d\mu\left(W^{\alpha\beta,a}\right) \exp\left(-\frac{\hat{q}_2}{2} ny + \frac{(\hat{q}_2 - \hat{q}_1)}{2}\sum_{\alpha\beta}\left(\sum_a W^{\alpha\beta,a}\right)^2\right) \times$$
$$\times \exp\left(\frac{(\hat{q}_1 - \hat{q}_0)}{2}\sum_{\alpha}\left(\sum_{\beta,a} W^{\alpha\beta,a}\right)^2 + \frac{\hat{q}_0}{2}\left(\sum_{\alpha\beta,a} W^{\alpha\beta,a}\right)^2 + \hat{S}\sum_{\alpha\beta,a} \tilde{W}^{\alpha\beta} W^{\alpha\beta,a}\right) \tag{A.2.14}$$

By means of a Hubbard-Stratonovich transformation

$$\exp\left(\frac{b}{2}x^2\right) = \int Dz \exp\left(t\sqrt{b}z\right)$$

on the quadratic term $\left(\sum_{\alpha\beta,a} W^{\alpha\beta,a}\right)^2$, the expression factorizes over the replica index α, thus obtaining:

$$G_S = e^{-\frac{\hat{q}_2}{2} ny} \int Dz_0 \left[\prod_{\beta} d\mu\left(\tilde{W}^{\beta}\right) \int \prod_{\beta,a} d\mu\left(W^{\beta,a}\right) \exp\left(\frac{(\hat{q}_2 - \hat{q}_1)}{2}\sum_{\beta}\left(\sum_a W^{\beta,a}\right)^2\right) \times\right.$$
$$\left. \times \exp\left(\frac{(\hat{q}_1 - \hat{q}_0)}{2}\left(\sum_{\beta,a} W^{\beta,a}\right)^2 + z_0\sqrt{\hat{q}_0}\sum_{\beta,a} W^{\beta,a} + \hat{S}\sum_{\beta,a} \tilde{W}^{\beta} W^{\beta,a}\right)\right]^{\frac{n}{m}} \tag{A.2.15}$$

Another transformation of the term $\left(\sum_{\beta,a} W^{\alpha\beta,a}\right)$ allows to factorize over the index β:

$$\begin{aligned} G_S &= e^{-\frac{\hat{q}_2}{2}ny}\int Dz_0\left\{\int Dz_1\left[\int d\mu\left(\tilde{W}\right)\int\prod_a d\mu\left(W^a\right)\exp\left(\frac{(\hat{q}_2-\hat{q}_1)}{2}\left(\sum_a W^a\right)^2\right)\times\right.\right. \\ &\left.\left.\times\exp\left(z_1\sqrt{\hat{q}_1-\hat{q}_0}\sum_a W^a + z_0\sqrt{\hat{q}_0}\sum_a W^a + \hat{S}\sum_a \tilde{W}W^a\right)\right]^m\right\}^{\frac{n}{m}} \end{aligned} \tag{A.2.16}$$

and again on the term $\left(\sum_a W^a\right)^2$, with a final factorization over the index a:

$$\begin{aligned} G_S &= e^{-\frac{\hat{q}_2}{2}ny}\int Dz_0\left\{\int Dz_1\left[\int Dz_2\int d\mu\left(\tilde{W}\right)\left(\int d\mu\left(W\right)\exp\left(z_2\sqrt{\hat{q}_2-\hat{q}_1}W\right)\times\right.\right.\right. \\ &\left.\left.\left.\times\exp\left(z_1\sqrt{\hat{q}_1-\hat{q}_0}W + z_0\sqrt{\hat{q}_0}W + \hat{S}\tilde{W}W\right)\right)^y\right]^m\right\}^{\frac{m}{n}} \end{aligned} \tag{A.2.17}$$

Let us then consider the specific case of binary variables $W, \tilde{W} \in \{-1, +1\}$ and perform the sum over W explicitly, thus obtaining:

$$G_S = e^{-\frac{\hat{q}_2}{2}ny}\int Dz_0\left\{\int Dz_1\left[\int Dz_2\sum_{\tilde{W}=\pm 1}\left(2\cosh\left(\tilde{A}\left(z_0,z_1,z_2;\tilde{W}\right)\right)\right)^y\right]^m\right\}^{\frac{n}{m}} \tag{A.2.18}$$

where

$$\tilde{A}\left(z_0,z_1,z_2;\tilde{W}\right) = z_2\sqrt{\hat{q}_2-\hat{q}_1} + z_1\sqrt{\hat{q}_1-\hat{q}_0} + z_0\sqrt{\hat{q}_0} + \hat{S}\tilde{W} \tag{A.2.19}$$

Performing the limit $n \to 0$ one obtains:

$$\frac{\log G_S}{n} = -\frac{\hat{q}_2}{2}y + \mathcal{G}_S \tag{A.2.20}$$

where

$$\mathcal{G}_S = \frac{1}{m}\int Dz_0 \log\left(\int Dz_1\left[\int Dz_2\sum_{\tilde{W}=\pm 1}\left(2\cosh\left(\tilde{A}\left(z_0,z_1,z_2;\tilde{W}\right)\right)\right)^y\right]^m\right) \tag{A.2.21}$$

A.2.2.4. *Energetic term.* Let us plug the 1-RSB Ansatz (A.2.13) in eq. (A.2.12) and get:

$$\begin{aligned} G_E &= \int\prod_{\alpha\beta,a}\frac{d\lambda^{\alpha\beta,a}d\hat{\lambda}^{\alpha\beta,a}}{2\pi}\prod_{\alpha\beta,a}\theta\left(\lambda^{\alpha\beta,a}\right)\exp\left(i\sum_{\alpha\beta,a}\lambda^{\alpha\beta,a}\hat{\lambda}^{\alpha\beta,a} - \frac{1}{2}\sum_{\alpha\beta,a}\left(\hat{\lambda}^{\alpha\beta,a}\right)^2\right)\times \\ &\times\exp\left(-q_2\sum_{\alpha\beta}\sum_{a>b}\hat{\lambda}^{\alpha\beta,a}\hat{\lambda}^{\alpha\beta,b} - q_1\sum_{\alpha,\beta>\beta'}\sum_{ab}\hat{\lambda}^{\alpha\beta,a}\hat{\lambda}^{\alpha\beta',b}\right)\times \\ &\times\exp\left(-q_0\sum_{\alpha>\alpha',\beta\beta'}\sum_{ab}\hat{\lambda}^{\alpha\beta,a}\hat{\lambda}^{\alpha'\beta',b}\right) \end{aligned} \tag{A.2.22}$$

With the use of the formula

$$\sum_{i>j} a_i a_j = \frac{1}{2}\left(\left(\sum_i a_i\right)^2 - \sum_i a_i^2\right)$$

for the various quadratic terms in λ's and $\hat{\lambda}'$s in eq. (A.2.22), one obtains:

$$\begin{aligned} G_E &= \int \prod_{\alpha\beta,a} \frac{d\lambda^{\alpha\beta,a} d\hat{\lambda}^{\alpha\beta,a}}{2\pi} \prod_{\alpha\beta,a} \theta\left(\lambda^{\alpha\beta,a}\right) \exp\left(i \sum_{\alpha\beta,a} \lambda^{\alpha\beta,a}\hat{\lambda}^{\alpha\beta,a} - \frac{1}{2}\sum_{\alpha\beta,a}\left(\hat{\lambda}^{\alpha\beta,a}\right)^2\right) \times \\ &\times \exp\left(-q_2 \sum_{\alpha\beta}\left(\left(\sum_a \hat{\lambda}^{\alpha\beta,a}\right)^2 - \sum_a \left(\hat{\lambda}^{\alpha\beta,a}\right)^2\right)\right) \times \\ &\times \exp\left(-q_1 \left(\sum_{\alpha}\left(\sum_{\beta,b} \hat{\lambda}^{\alpha\beta,a}\right)^2 - \sum_{\beta,b}\left(\hat{\lambda}^{\alpha\beta,a}\right)^2\right)\right) \times \\ &\times \exp\left(-q_0 \left(\left(\sum_{\alpha\beta,a} \hat{\lambda}^{\alpha\beta,a}\right)^2 - \sum_{\alpha}\left(\sum_{\beta,a} \hat{\lambda}^{\alpha\beta,a}\right)^2\right)\right) \end{aligned} \tag{A.2.23}$$

Let us then linearize the term multiplying q_0 by means of a Hubbard-Stratonovich transformation, thereby factorizing over the replica index α:

$$\begin{aligned} G_E &= \int Dz_0 \left[\prod \frac{d\lambda^{\beta,a} d\hat{\lambda}^{\beta,a}}{2\pi} \prod_{\beta,a} \theta\left(\lambda^{\beta,a}\right) \exp\left(i \sum_{\beta,a} \lambda^{\beta,a}\hat{\lambda}^{\beta,a} - \frac{(1-q_2)}{2}\sum_{\beta,a}\left(\hat{\lambda}^{\beta,a}\right)^2\right) \times \right. \\ &\times \exp\left(-\frac{(q_2-q_1)}{2}\sum_{\beta}\left(\sum_a \hat{\lambda}^{\beta,a}\right)^2 - \frac{(q_1-q_0)}{2}\left(\sum_{\beta,a} \hat{\lambda}^{\beta,a}\right)^2\right) \times \\ &\left. \times \exp\left(-iz_0\sqrt{q_0}\sum_{\beta,a} \hat{\lambda}^{\beta,a}\right)\right]^{\frac{n}{m}} \end{aligned} \tag{A.2.24}$$

Performing two more Hubbard-Stratonovich transformations allows to factorize over the relevant indices β and a:

$$G_E = \int Dz_0 \left\{\int Dz_1 \left[\int Dz_2\, H\left(A\left(z_0, z_1, z_2\right)\right)^y\right]^m\right\}^{\frac{n}{m}} \tag{A.2.25}$$

where

$$A\left(z_0, z_1, z_2\right) = \frac{z_0\sqrt{q_0} + z_1\sqrt{q_1 - q_0} + z_2\sqrt{q_2 - q_1}}{\sqrt{1-q_2}} \tag{A.2.26}$$

and the the gaussian integral over $\hat{\lambda}$ has been performed, writing the definite gaussian integral over λ as an H function, $H(x) = \int_x^{+\infty} Dz$. In the limit $n \to 0$ we get:

$$\mathcal{G}_E = \frac{\log G_E}{n} = \frac{1}{m} \int Dz_0 \log \left(\int Dz_1 \left[\int Dz_2 \, H\left(A\left(z_0, z_1, z_2 \right) \right)^y \right]^m \right) \tag{A.2.27}$$

A.2.2.5. *Final* 1-*RSB expression.* Plugging eqs. (A.2.21) and (A.2.27) into eq. (A.2.10), one obtains:

$$\langle \Omega(S,y)^n \rangle_{\{\xi^\mu\}} = \exp\left(-Nny \mathscr{F}(S,y) \right)$$

so that the eq. (A.2.4) is:

$$\mathscr{F}(S,y) = -\left(y\frac{m}{2} q_0 \hat{q}_0 - y\frac{(m-1)}{2} q_1 \hat{q}_1 - \frac{(y-1)}{2} q_2 \hat{q}_2 - \frac{\hat{q}_2}{2} - S\hat{S} + \frac{1}{y}\mathcal{G}_S + \frac{\alpha}{y}\mathcal{G}_E \right) \tag{A.2.28}$$

The order parameters are obtained by the saddle point equations. In order to study the zero-temperature limit $y \to \infty$, it is convenient to rearrange the terms as:

$$\mathscr{F}(S,y) = -\left(\frac{my}{2}\left(q_0\hat{q}_0 - q_1\hat{q}_1 \right) - \frac{y}{2}\left(q_2\hat{q}_2 - q_1\hat{q}_1 \right) - \frac{\hat{q}_2}{2}\left(1 - q_2 \right) - S\hat{S} + \frac{1}{y}\mathcal{G}_S + \frac{\alpha}{y}\mathcal{G}_E \right) \tag{A.2.29}$$

from which one sees that in this limit the parameters must scale as:

$$\begin{aligned} m &\to \frac{x}{y} \\ q_2 &\to q_1 + \frac{\delta q}{y} \\ \hat{q}_2 &\to \hat{q}_1 + \frac{\delta \hat{q}}{y} \end{aligned}$$

giving, to the leading order in y:

$$\mathscr{F}(S,\infty) = -\left(\frac{x}{2}\left(q_0\hat{q}_0 - q_1\hat{q}_1 \right) - \frac{1}{2}\left(q_1\delta\hat{q} + \hat{q}_1\delta q \right) - \frac{\hat{q}_1}{2}\left(1 - q_1 \right) - S\hat{S} + \mathcal{G}_S^\infty + \alpha\mathcal{G}_E^\infty \right) \tag{A.2.30}$$

where

$$\mathcal{G}_S^\infty = \frac{1}{x} \int Dz_0 \log \left(\int Dz_1 \, e^{x\,\tilde{B}(z_0,z_1)} \right) \tag{A.2.31}$$

$$\tilde{B}(z_0,z_1) = \max_{z_2 \in \mathbb{R}, \tilde{W} = \pm 1} \left(\tilde{A}^\infty \left(z_0, z_1, z_2; \tilde{W} \right) \right) \tag{A.2.32}$$

$$\tilde{A}^\infty \left(z_0, z_1, z_2; \tilde{W} \right) = -\frac{z_2^2}{2} + \log\left(2\cosh\left(z_2\sqrt{\delta\hat{q}} + z_1\sqrt{\hat{q}_1 - \hat{q}_0} + z_0\sqrt{\hat{q}_0} + \hat{S}\tilde{W} \right) \right) \tag{A.2.33}$$

$$\mathcal{G}_E^\infty = \frac{1}{x} \int Dz_0 \log \left(\int Dz_1 \, e^{x\,B(z_0,z_1)} \right) \tag{A.2.34}$$

$$B(z_0,z_1) = \max_{z_2} \left(A^\infty(z_0,z_1,z_2) \right) \tag{A.2.35}$$

$$A^\infty(z_0,z_1,z_2) = -\frac{z_2^2}{2} + \log\left(H\left(\frac{z_0\sqrt{q_0} + z_1\sqrt{q_1 - q_0} + z_2\sqrt{\delta q}}{\sqrt{1 - q_1}} \right) \right) \tag{A.2.36}$$

The expressions for $\mathcal{G}_S^\infty$ and $\mathcal{G}_E^\infty$ are obtained by the saddle point method, using $y \to \infty$. The resulting saddle point equations for the order parameters are:

$$\hat{q}_0 = \frac{\alpha}{x\sqrt{\delta q}} \int Dz_0 \frac{\int Dz_1\, e^{x\, B(z_0,z_1)}\, z_E^\star(z_0,z_1) \left(\frac{z_1}{\sqrt{q_1-q_0}} - \frac{z_0}{\sqrt{q_0}} \right)}{\int Dz_1\, e^{x\, B(z_0,z_1)}} \tag{A.2.37}$$

$$\hat{q}_1 = \frac{\alpha}{\delta q} \int Dz_0 \frac{\int Dz_1\, e^{x\, B(z_0,z_1)}\, \left(z_E^\star(z_0,z_1)\right)^2}{\int Dz_1\, e^{x\, B(z_0,z_1)}} \tag{A.2.38}$$

$$\delta\hat{q} = (1-x)\,\hat{q}_1 + \frac{\alpha}{\sqrt{\delta q}} \int Dz_0 \frac{\int Dz_1\, e^{x\, B(z_0,z_1)} \left(z_E^\star(z_0,z_1) \frac{z_1}{\sqrt{q_1-q_0}} + \frac{b(z_0,z_1)}{\sqrt{\delta q}} \right)}{\int Dz_1\, e^{x\, B(z_0,z_1)}} \tag{A.2.39}$$

$$q_0 = \frac{1}{x\sqrt{\delta\hat{q}}} \int Dz_0 \frac{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}\, z_S^\star(z_0,z_1) \left(\frac{z_1}{\sqrt{\hat{q}_1-\hat{q}_0}} - \frac{z_0}{\sqrt{\hat{q}_0}} \right)}{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}} \tag{A.2.40}$$

$$q_1 = \frac{1}{\delta\hat{q}} \int Dz_0 \frac{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}\, \left(z_S^\star(z_0,z_1)\right)^2}{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}} \tag{A.2.41}$$

$$\delta q = (1-x)\,q_1 - 1 + \frac{1}{\sqrt{\delta\hat{q}}} \int Dz_0 \frac{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}\, z_S^\star(z_0,z_1) \left(\frac{z_1}{\sqrt{\hat{q}_1-\hat{q}_0}} \right)}{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}} \tag{A.2.42}$$

$$S = \frac{1}{\sqrt{\delta\hat{q}}} \int Dz_0 \frac{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}\, z_S^\star(z_0,z_1)\, \mathrm{sign}\left(\hat{S}\left(z_1\sqrt{\hat{q}_1-\hat{q}_0} + z_0\sqrt{\hat{q}_0} \right) \right)}{\int Dz_1\, e^{x\, \tilde{B}(z_0,z_1)}} \tag{A.2.43}$$

where

$$z_S^\star(z_0,z_1) = \mathrm{argmax}_{z_2\in\mathbb{R}} \left(\max_{\tilde{W}=\pm 1} \left(\tilde{A}^\infty\left(z_0,z_1,z_2;\tilde{W}\right) \right) \right) \tag{A.2.44}$$

$$z_E^\star(z_0,z_1) = \mathrm{argmax}_{z_2\in\mathbb{R}} \left(A^\infty(z_0,z_1,z_2) \right) \tag{A.2.45}$$

$$b(z_0,z_1) = \frac{z_E^\star(z_0,z_1)\sqrt{\delta q}}{1-q_1} \left(z_0\sqrt{q_0} + z_1\sqrt{q_1-q_0} + z_E^\star(z_0,z_1)\sqrt{\delta q} \right) \tag{A.2.46}$$

The parameter x is implicitly set by the equation:

$$\frac{1}{2}\left(q_0\hat{q}_0 - q_1\hat{q}_1\right) + \frac{\partial\mathcal{G}_S^\infty}{\partial x} + \alpha\frac{\partial\mathcal{G}_E^\infty}{\partial x} = 0 \tag{A.2.47}$$

where

$$\frac{\partial\mathcal{G}_S^\infty}{\partial x} = \frac{1}{x}\int Dz_0 \frac{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)}\, \tilde{B}(z_0,z_1)}{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)}} - \frac{\mathcal{G}_S^\infty}{x} \tag{A.2.48}$$

$$\frac{\partial\mathcal{G}_E^\infty}{\partial x} = \frac{1}{x}\int Dz_0 \frac{\int Dz_1\, e^{x\,B(z_0,z_1)}\, B(z_0,z_1)}{\int Dz_1\, e^{x\,B(z_0,z_1)}} - \frac{\mathcal{G}_E^\infty}{x} \tag{A.2.49}$$

In this limit, and since the goal is to optimize over x, $-\mathscr{F}(S,\infty)$ is equal to the *local entropy* $\mathscr{S}_I$, i.e. the entropy of the solutions W (which has formally the role of an energy in our model). This is shown in Fig. 5.2.1. The *external entropy*, i.e. the entropy of the reference configurations $\tilde{W}$, has two components in the 1-RSB scenario. The first, usually called *complexity*, accounts for the number of clusters of $\tilde{W}$, and is set to zero by optimizing $\mathscr{F}$ over x (see above). The second is computed from a

first-order expansion in y, since $\mathscr{F}(S,y) = -\mathscr{S}_I - \frac{1}{y}\mathscr{S}_E$, giving:

$$\mathscr{S}_E = \frac{1}{2}\left(-\delta\hat{q} - \delta q\delta\hat{q} + \delta\hat{q}q_1 + \delta q\hat{q}_1\right) + \mathcal{C}_S^{\infty} + \alpha\mathcal{C}_E^{\infty} \tag{A.2.50}$$

$$\mathcal{C}_S^{\infty} = -\frac{1}{2}\int Dz_0 \frac{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)} \log\left(1 - \delta\hat{q} + z_S^{\star}(z_0,z_1)^2\right)}{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)}} \tag{A.2.51}$$

$$\mathcal{C}_E^{\infty} = -\frac{1}{2}\int Dz_0 \frac{\int Dz_1\, e^{x\,B(z_0,z_1)} \left(\log\left(1 + z_E^{\star}(z_0,z_1)^2 + b(z_0,z_1)\right) - b(z_0,z_1)\right)}{\int Dz_1\, e^{x\,B(z_0,z_1)}} \tag{A.2.52}$$

A.2.3. Reference configurations energy and constrained case.

A.2.3.1. *Breaking the symmetry over reference configurations.* The expression of eq. (A.2.28) does not involve any overlaps relative to the reference configurations. However, in order to compute the average energy associated with the reference configurations, it is necessary to obtain such quantities (see below). To extract the order parameter this end, it is useful to introduce a modified free energy:

$$\begin{aligned}\mathscr{F}_C(S,y) &= -\frac{1}{Ny}\left\langle \log\left(\Omega_C(S,y)\right)\right\rangle_{\{\xi^\mu\}} \\ &= -\frac{1}{Ny}\left\langle f\left(\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i\xi_i^\mu\right)\log\left(\sum_{\{\tilde{W}\}}\mathcal{N}_\xi\left(\tilde{W},S\right)^y\right)\right\rangle_{\{\xi^\mu\}}\end{aligned} \tag{A.2.53}$$

i.e. one in which an additional term $f\left(\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i\xi_i^\mu\right)$ was included in the expression. Setting $f(x) = \Theta(x)$ one recovers the case in which the reference solutions $\tilde{W}$ are also constrained to be solutions to the classification problem (chapter 4); on the other hand, with the introduction of a parameter η such that $\lim_{\eta\to 0} f(x) = 1$, the previous case is recovered in the limit $\eta \to 0$, i.e. this amounts to introduce a symmetry-breaking term and then making it vanish at the end of the computation.

The computation follows along the lines of the previous case, but requires the introduction of some additional order parameters: $\tilde{q}^{\alpha\beta;\alpha'\beta'} = \frac{1}{\sqrt{N}}\tilde{W}^{\alpha\beta}\cdot\tilde{W}^{\alpha'\beta'}$ for the overlaps between reference configurations and $S^{\alpha'\beta';\alpha\beta a} = \frac{1}{\sqrt{N}}\tilde{W}^{\alpha'\beta'}\cdot W^{\alpha\beta;a}$ for the overlaps between reference configurations and solutions. With the 1-RSB Ansatz, and including the constraint on the overlaps, the order parameters take the form:

$$\tilde{q}^{\alpha\beta;\alpha'\beta'} = \begin{cases} 1 & \text{if } \alpha = \alpha', \beta = \beta' \\ \tilde{q}_1 & \text{if } \alpha = \alpha', \beta \neq \beta' \\ \tilde{q}_0 & \text{if } \alpha \neq \alpha' \end{cases} \tag{A.2.54}$$

$$S^{\alpha'\beta';\alpha\beta a} = \begin{cases} S & \text{if } \alpha = \alpha', \beta = \beta' \\ \tilde{S}_1 & \text{if } \alpha = \alpha', \beta \neq \beta' \\ \tilde{S}_0 & \text{if } \alpha \neq \alpha' \end{cases}$$

and analogous expressions for the conjugate parameters. The final expression for the free energy is:

$$\begin{aligned}\mathscr{F}_C(S,y) &= -\left(\frac{m}{2y}\left(\tilde{q}_0\hat{\tilde{q}}_0-\tilde{q}_1\hat{\tilde{q}}_1\right)-\frac{1}{2y}\hat{\tilde{q}}_1\left(1-\tilde{q}_1\right)+\frac{my}{2}\left(q_0\hat{q}_0-q_1\hat{q}_1\right)+\right.\\ &\quad -\frac{y}{2}\left(q_2\hat{q}_2-q_1\hat{q}_1\right)-\frac{\hat{q}_2}{2}\left(1-q_2\right)-\left(S\hat{S}-\tilde{S}_1\hat{\tilde{S}}_1\right)+\\ &\quad \left.+m\left(\tilde{S}_0\hat{\tilde{S}}_0-\tilde{S}_1\hat{\tilde{S}}_1\right)+\frac{1}{y}\mathcal{G}_{CS}+\frac{\alpha}{y}\mathcal{G}_{CE}\right) \qquad \text{(A.2.55)}\end{aligned}$$

where

$$\begin{aligned}\mathcal{G}_{CS} &= \frac{1}{m}\int D\tilde{z}_0\int Dz_0\log\left(\int D\tilde{z}_1\int Dz_1\left[\sum_{\tilde{W}=\pm1}e^{\tilde{W}\,K(\tilde{z}_0,z_0,\tilde{z}_1,z_1)}\times\right.\right. \qquad \text{(A.2.56)}\\ &\quad \left.\left.\times\int Dz_2\left(2\cosh\left(\tilde{A}_C\left(z_0,z_1,z_2;\tilde{W}\right)\right)\right)^y\right]^m\right)\\ \tilde{A}_C\left(z_0,z_1,z_2;\tilde{W}\right) &= z_2\sqrt{\hat{q}_2-\hat{q}_1}+z_1\sqrt{\hat{q}_1-\hat{q}_0}+z_0\sqrt{\hat{q}_0}+\left(\hat{S}-\hat{\tilde{S}}_1\right)\tilde{W} \qquad \text{(A.2.57)}\\ K\left(\tilde{z}_0,z_0,\tilde{z}_1,z_1\right) &= \tilde{z}_1\sqrt{\left(\hat{\tilde{q}}_1-\hat{\tilde{q}}_0\right)-\frac{\left(\hat{\tilde{S}}_1-\hat{\tilde{S}}_0\right)^2}{\hat{q}_1-\hat{q}_0}}+z_1\frac{\hat{\tilde{S}}_1-\hat{\tilde{S}}_0}{\sqrt{\hat{q}_1-\hat{q}_0}}+ \qquad \text{(A.2.58)}\\ &\quad +\tilde{z}_0\sqrt{\hat{\tilde{q}}_0-\frac{\left(\hat{\tilde{S}}_0\right)^2}{\hat{q}_0}}+z_0\frac{\hat{\tilde{S}}_0}{\sqrt{\hat{q}_0}}\\ \mathcal{G}_{CE} &= \frac{1}{m}\int D\tilde{z}_0\int Dz_0\log\left(\int D\tilde{z}_1\int Dz_1\left[\int Dz_2 H\left(A\left(z_0,z_1,z_2\right)^y\right)\times\right.\right. \qquad \text{(A.2.59)}\\ &\quad \left.\left.\times\ L\left(\tilde{z}_0,z_0,\tilde{z}_1,z_1,z_2\right)\right]^m\right)\\ L\left(\tilde{z}_0,z_0,\tilde{z}_1,z_1,z_2\right) &= \int D\tilde{\lambda}\,f\left(\frac{z_2\frac{S-\tilde{S}_1}{\sqrt{1-\tilde{q}_1}}+\tilde{\lambda}\sqrt{\left(q_2-q_1\right)-\frac{\left(S-\tilde{S}_1\right)^2}{1-\tilde{q}_1}}}{\sqrt{q_2-q_1}}+z_1\frac{\tilde{S}_1-\tilde{S}_0}{\sqrt{q_1-q_0}}+\right.\\ &\quad \left.+z_0\frac{\tilde{S}_0}{\sqrt{q_0}}+\tilde{z}_1\sqrt{\left(\tilde{q}_1-\tilde{q}_0\right)-\frac{\left(\tilde{S}_1-\tilde{S}_0\right)^2}{q_1-q_0}}+\tilde{z}_0\sqrt{\tilde{q}_0-\frac{\left(\tilde{S}_0\right)^2}{q_0}}\right) \qquad \text{(A.2.60)}\end{aligned}$$

From these expression, one immediately notes that the dependency on the function f only enters the equations through the expression of L in $\mathcal{G}_{CE}$. Also, this expression does not depend on y. This has two consequences:

(1) In the case where $\lim_{\eta\to0} f(x) = 1$, $\mathcal{G}_{CE} \to \mathcal{G}_E$, i.e. the expression does not depend any more on $\tilde{q}_1$, $\tilde{q}_0$, $\tilde{S}_1$ or $\tilde{S}_0$ and simplifies to the previous case, as expected. In turn, this implies that the conjugated order parameters $\hat{\tilde{q}}_1$, $\hat{\tilde{q}}_0$, $\hat{\tilde{S}}_1$ and $\hat{\tilde{S}}_0$ all tend to 0, thus reducing the expression of the free energy to the previous case eq. (A.2.28);
(2) In the limit $y \to \infty$ in the constrained case ($f(x) = \Theta(x)$), one gets $\mathcal{G}_{CE} \to \mathcal{G}_E$, since the term with y in the exponent dominates the saddle point expansion. Again , the expression (A.2.28)

is recovered for the free energy of the system, which means that the local entropy is unchanged in the constrained case. This may suggest that, even in the unconstrained case, the reference configuration $\tilde{W}$ is never "too far" from an actual solution to the problem (more precisely, within a distance $o(N)$ from a solution). Note, however, that — as one would expect — the external entropy is different in this case, since it depends on the first order expansion in y, which is affected by the L term.

Following observation 1, the expression for the order parameters $\tilde{q}_1$, $\tilde{q}_0$, $\tilde{S}_1$ and $\tilde{S}_0$ may be derived by using the saddle point equations and by assuming that the conjugate parameters $\hat{\tilde{q}}_1$, $\hat{\tilde{q}}_0$, $\hat{\tilde{S}}_1$ and $\hat{\tilde{S}}_0$ are of order $\eta \ll 1$ and taking the leading order in the resulting expression. The only results needed are the one for $\tilde{S}_1$ and $\tilde{S}_0$, which turn out to be:

$$\tilde{S}_1 = \frac{S}{1-m} + \frac{1}{y(m-1)\sqrt{\hat{q}_1 - \hat{q}_0}} \int Dz_0 \frac{I_{dz}(z_0)}{I_s(z_0)} \tag{A.2.61}$$

$$\tilde{S}_0 = \frac{1}{my}\left(\frac{1}{\sqrt{\hat{q}_1 - \hat{q}_0}} \int Dz_0 \frac{I_{dz}(z_0)}{I_s(z_0)} - \frac{1}{\sqrt{\hat{q}_0}} \int Dz_0\, z_0 \frac{I_d(z_0)}{I_s(z_0)} \right) \tag{A.2.62}$$

$$I_s(z_0) = \int Dz_1 \left[\int Dz_2 \sum_{\tilde{W}=\pm 1} \left(2\cosh\left(\tilde{A}\left(z_0, z_1, z_2; \tilde{W} \right) \right) \right)^y \right]^m \tag{A.2.63}$$

$$I_d(z_0) = \int Dz_1 \left[\int Dz_2 \sum_{\tilde{W}=\pm 1} \left(2\cosh\left(\tilde{A}\left(z_0, z_1, z_2; \tilde{W} \right) \right) \right)^y \right]^{m-1} \times \tag{A.2.64}$$
$$\times \left(\int Dz_2 \sum_{\tilde{W}=\pm 1} \tilde{W} \left(2\cosh\left(\tilde{A}\left(z_0, z_1, z_2; \tilde{W} \right) \right) \right)^y \right)$$

$$I_{dz}(z_0) = \int Dz_1\, z_1 \left[\int Dz_2 \sum_{\tilde{W}=\pm 1} \left(2\cosh\left(\tilde{A}\left(z_0, z_1, z_2; \tilde{W} \right) \right) \right)^y \right]^{m-1} \times \tag{A.2.65}$$
$$\times \left(\int Dz_2 \sum_{\tilde{W}=\pm 1} \tilde{W} \left(2\cosh\left(\tilde{A}\left(z_0, z_1, z_2; \tilde{W} \right) \right) \right)^y \right)$$

In the limit $y \to \infty$, the following scaling holds:

$$\tilde{S}_1 = S - \frac{\delta S}{y}$$

and finally the expressions:

$$\delta S = \frac{1}{\sqrt{\hat{q}_1 - \hat{q}_0}} \int Dz_0 \frac{\int Dz_1\, z_1 e^{x\,\tilde{B}(z_0,z_1)} \tilde{W}^\star(z_0, z_1)}{\int Dz_1 e^{x\,\tilde{B}(z_0,z_1)}} - xS \tag{A.2.66}$$

$$\tilde{S}_0 = \frac{1}{x\sqrt{\hat{q}_1 - \hat{q}_0}} \int Dz_0 \frac{\int Dz_1\, z_1 e^{x\,\tilde{B}(z_0,z_1)} \tilde{W}^\star(z_0, z_1)}{\int Dz_1 e^{x\,\tilde{B}(z_0,z_1)}} + \tag{A.2.67}$$
$$- \frac{1}{x\sqrt{\hat{q}_0}} \int Dz_0\, z_0 \frac{\int Dz_1\, e^{x\,\tilde{B}(z_0,z_1)} \tilde{W}^\star(z_0, z_1)}{\int Dz_1 e^{x\,\tilde{B}(z_0,z_1)}}$$

where

$$\tilde{W}^{\star}\left(z_0, z_1\right) = \operatorname{argmax}_{\tilde{W}=\pm 1}\left(\max_{z_2\in\mathbb{R}}\left(\tilde{A}^{\infty}\left(z_0, z_1, z_2; \tilde{W}\right)\right)\right) \tag{A.2.68}$$

A.2.3.2. *Energy density.* The typical energy density of the unconstrained reference configurations $\tilde{W}$ gives the information related to the probability of classifying incorrectly a pattern $\xi^{\star}$ drawn at random from the training set. This probability can be obtained by calculating:

$$P\left(\sigma^{\star} \neq 1\right) = \left\langle \Theta\left(-\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i \xi_i^{\star}\right)\right\rangle_{\tilde{W}} \tag{A.2.69}$$

where the average is defined over the weighted measure $d\mu_W\left(\tilde{W}\right) = d\mu\left(\tilde{W}\right)\mathcal{N}_{\xi}\left(\tilde{W}, S\right)^{y}$. Since:

$$\left\langle \prod_{\mu}\Theta\left(-\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i \xi_i^{\mu}\right)\right\rangle_{\tilde{W}} = \frac{\int d\mu_W\left(\tilde{W}\right)\prod_{\mu}\Theta\left(-\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i\xi_i^{\mu}\right)}{\int d\mu_W\left(\tilde{W}\right)} \tag{A.2.70}$$

this calculation can be carried out straightforwardly by exploiting the replica trick, i.e. by rewriting the ratio in (A.2.70) as:

$$\begin{aligned}&\lim_{n\to 0}\int d\mu_W\left(\tilde{W}\right)\Theta\left(-\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i\xi_i^{\star}\right)\left(\int d\mu_W\left(\tilde{W}\right)\right)^{n-1} = \\ &= \lim_{n\to 0}\int\prod_c d\mu_W\left(\tilde{W}^c\right)\Theta\left(-\frac{1}{\sqrt{N}}\sum_i \tilde{W}_i^1\xi_i^{\star}\right)\end{aligned} \tag{A.2.71}$$

where one introduces $n-1$ unconstrained replicas of the reference solution, leaving out the replica index 1 for the $\tilde{W}$-replica coupled to the pattern $\xi^{\star}$ by the constraint. In this way the quenched disorder can be averaged out, and in the $n \to 0$ limit one recovers the initial expression.

As noted in the previous section, when one extracts the overlaps referred to the reference configurations by introducing vanishing constraints (i.e. when $\eta \to 0$), the conjugate parameters related to these overlaps tend to vanish as well.

Therefore, if one organizes the calculation similarly to the previous ones, it is easy to see that in (A.2.71) the entropic terms cancel out and the only non-zero contribution to the average comes from the energetic part G_E', where:

$$
\begin{aligned}
G'_E &= \int \frac{d\tilde{\lambda}^1 d\hat{\tilde{\lambda}}^1}{2\pi} \prod_{\alpha\beta,a} \frac{d\lambda^{\alpha\beta,a} d\hat{\lambda}^{\alpha\beta,a}}{2\pi} \Theta\left(-\tilde{\lambda}^1\right) \prod_{\alpha\beta,a} \Theta\left(\lambda^{\alpha\beta,a}\right) \times \\
&\times \exp\left(i \left(\tilde{\lambda}^1 \hat{\tilde{\lambda}}^1 + \sum_{\alpha\beta,a} \lambda^{\alpha\beta,a} \hat{\lambda}^{\alpha\beta,a} \right) - \frac{1}{2} \left(\left(\hat{\tilde{\lambda}}^1\right)^2 + \sum_{\alpha\beta,a} \left(\hat{\lambda}^{\alpha\beta,a}\right)^2 \right) \right) \times \\
&\times \exp\left(-q_2 \sum_{\alpha\beta} \sum_{a>b} \hat{\lambda}^{\alpha\beta,a} \hat{\lambda}^{\alpha\beta,b} - q_1 \sum_{\alpha,\beta>\beta'} \sum_{ab} \hat{\lambda}^{\alpha\beta,a} \hat{\lambda}^{\alpha\beta',b} \right) \times \\
&\times \exp\left(-q_0 \sum_{\alpha>\alpha',\beta\beta'} \sum_{ab} \hat{\lambda}^{\alpha\beta,a} \hat{\lambda}^{\alpha'\beta',b} - \sum_{a} \hat{\tilde{\lambda}}^1 \hat{\lambda}^{11,a} \left(S - \tilde{S}_1\right) \right) \times \\
&\times \exp\left(-\sum_{\beta} \sum_{a} \hat{\tilde{\lambda}}^1 \hat{\lambda}^{1\beta,a} \left(\tilde{S}_1 - \tilde{S}_0\right) - \sum_{a\beta} \sum_{a} \hat{\tilde{\lambda}}^1 \hat{\lambda}^{\alpha\beta,a} \tilde{S}_0 \right) \qquad \text{(A.2.72)}
\end{aligned}
$$

The final expression is the following:

$$
\begin{aligned}
&P\left(\sigma^\star \neq 1\right) = \qquad \text{(A.2.73)} \\
&= \int Dz_0 \frac{\int Dz_1 \left(\int Dz_2 H\left(A\left(z_0,z_1,z_2\right)\right)^y \right)^{m-1} \int Dz_2 H\left(A\left(z_0,z_1,z_2\right)\right)^y H\left(-C\left(z_0,z_1,z_2\right)\right)}{\int Dz_1 \left(\int Dz_2 H\left(A\left(z_0,z_1,z_2\right)\right)^y \right)^m}
\end{aligned}
$$

with the definition (A.2.26), and where:

$$
C\left(z_0,z_1,z_2\right) = \frac{z_0 \frac{\tilde{S}_0}{\sqrt{q_0}} + z_1 \frac{\tilde{S}_1 - \tilde{S}_0}{\sqrt{q_1 - q_0}} + z_2 \frac{S - \tilde{S}_1}{\sqrt{q_2 - q_1}}}{\sqrt{1 - \frac{\tilde{S}_0^2}{q_0} - \frac{\left(\tilde{S}_1 - \tilde{S}_0\right)^2}{q_1 - q_0} - \frac{\left(S - \tilde{S}_1\right)^2}{q_2 - q_1}}}
$$

In the limit $y \to \infty$, Eq. A.2.73 simplifies to:

$$
P\left(\sigma^\star \neq 1\right) = \int Dz_0 \frac{\int Dz_1 e^{x\, B(z_0,z_1)} H\left(- \frac{z_0 \frac{\tilde{S}_0}{\sqrt{q_0}} + z_1 \frac{\tilde{S}_1 - \tilde{S}_0}{\sqrt{q_1 - q_0}} + z_2 \frac{\delta S}{\sqrt{\delta q}}}{\sqrt{1 - \frac{\tilde{S}_0^2}{q_0} - \frac{\left(\tilde{S}_1 - \tilde{S}_0\right)^2}{q_1 - q_0}}} \right)}{\int Dz_1 e^{x\, B(z_0,z_1)}} \qquad \text{(A.2.74)}
$$

which is shown in Fig. 5.2.2.

A.3. Heuristic Algorithm

Algorithm 1 Heuristic EdMC

Input: problem sample; parameters $t_{\max}$, t_{step}, y, γ, f_y and f_γ
Randomly initialize $\tilde{x}_i^0$
Alternatively, run BP with $\gamma = 0$ and set $\tilde{x}_i^0 = \text{sign}(h_i)$
Run BP with external fields $\gamma\tilde{x}_i^0$
Compute free energy F^0 from BP fixed point ($\bar{F}^0$ if BP does not converge)
while $t \leq t_{\max}$ **do**
  Retrieve fields h_i^t ($\bar{h}_i^t$ if BP did not converge)
  for $i = 1$ **to** N **do**
    $\Delta_i \leftarrow \tilde{x}_i^t\,(\gamma\tilde{x}_i^t - h_i^t)$
  end for
  Collect $V = \{i \mid \Delta_i > 0\}$ and sort it in descending order of Δ_i
  accepted $\leftarrow$ FALSE
  while NOT *accepted* **do**
    Propose a flip of the $\tilde{x}_i^t$ for all $i \in V$, producing $\tilde{x}^{t+1}$
    Run BP with new proposed external fields $\gamma\tilde{x}_i^{t+1}$
    Compute free energy F^{t+1} from BP fixed point ($\bar{F}^{t+1}$ if BP does not converge)
    with probability $e^{y\left(F^{t+1}-F^t\right)}$ **do** *accepted* $\leftarrow$ TRUE
    if NOT *accepted* **then**
      Remove the last element from V
      if $|V| = 0$ **then**
        exit and run EdMC with $\tilde{x}^t$ as initial configuration
      end if
    end if
  end while
  $t \leftarrow t + 1$
  Compute energy E of configuration $\tilde{x}^t$
  if $E = 0$ **then**
    retrieve solution $\tilde{x}^* = \tilde{x}^t$ and exit
  end if
  if $t \equiv 0 \pmod{t_{\text{step}}}$ **then**
    Annealing: $y \leftarrow y \times f_y$
    Scoping: $\gamma \leftarrow \gamma \times f_\gamma$ (run BP and update F^t)
  end if
end while

APPENDIX B

Inverse Dynamics: the patient zero problem

B.1. Efficient BP updates

An efficient form for the update equations of the ψ_i factor nodes is the following:

$$
\begin{aligned}
p_{\psi_i \to j}\left(t_i^{(j)}, t_{ji}, g_i^{(j)}\right) \quad &\propto \quad \sum_{g_i, t_i} \sum_{\left\{t_i^{(k)}, t_{ki}, g_i^{(k)}\right\}} m_{i\to\psi_i}\left(t_i, g_i\right) \times \\
&\quad \times \prod_{k\in\partial i\setminus j} m_{k\to\psi_i}\left(t_i^{(k)}, t_{ki}, g_i^{(k)}\right) \psi_i\left(t_i, g_i, \left\{\left(t_i^{(k)}, t_{ki}, g_i^{(k)}\right)\right\}_{k\in\partial i}\right) \\
&\propto \quad m_{i\to\psi_i}\left(t_i^{(j)}, g_i^{(j)}\right) \sum_{t_{ki}} \prod_{k\in\partial i\setminus j} m_{k\to\psi_i}\left(t_i^{(j)}, t_{ki}, g_i^{(j)}\right) \times \qquad \text{(B.1.1)} \\
&\quad \times \left[\delta\left(t_i^{(j)}, 0\right) + \delta\left(t_i^{(j)}, \left(1 + \min_{k\in\partial i}\left\{t_{ki}\right\}\right)\right)\right] \\
&\propto \quad \delta\left(t_i^{(j)}, 0\right) m_{i\to\psi_i}\left(0, g_i^{(j)}\right) \prod_{k\in\partial i\setminus j} \sum_{t_{ki}} m_{k\to\psi_i}\left(0, t_{ki}, g_i^{(j)}\right) + \qquad \text{(B.1.2)} \\
&\quad + m_{i\to\psi_i}\left(t_i^{(j)}, g_i^{(j)}\right) \mathbb{I}\left(t_i^{(j)} \le t_{ji} + 1\right) \prod_{k\in\partial i\setminus j} \sum_{t_{ki}\ge t_i^{(j)}-1} m_{k\to\psi_i}\left(t_i^{(j)}, t_{ki}, g_i^{(j)}\right) \\
&\quad - m_{i\to\psi_i}\left(t_i^{(j)}, g_i^{(j)}\right) \mathbb{I}\left(t_i^{(j)} < t_{ji} + 1\right) \prod_{k\in\partial i\setminus j} \sum_{t_{ki} > t_i^{(j)}-1} m_{k\to\psi_i}\left(t_i^{(j)}, t_{ki}, g_i^{(j)}\right)
\end{aligned}
$$

where in (B.1.2) we use the fact that

$$
\delta\left(t_i, \left(1 + \min_{j\in\partial i}\left\{t_{ji}\right\}\right)\right) = \prod_{j\in\partial i} \mathbb{I}\left(t_i \le t_{ji} + 1\right) - \prod_{j\in\partial i} \mathbb{I}\left(t_i < t_{ji} + 1\right).
$$

In order to switch to the simplified representation with σ_{ji}, σ_{ij} variables defined in (6.2.5) instead of t_{ji}, t_{ij} ones, one proceeds as follows. In equation (B.1.2) the sums over different configurations of $\left(t_{ki}, t_i^{(j)}\right)$ may be easily grouped together so that:

$$
\begin{aligned}
p_{\psi_i \to j}\left(t_i^{(j)}, \sigma_{ji}, g_i^{(j)}\right) \quad &\propto \quad \delta\left(t_i^{(j)}, 0\right) m_{i\to\psi_i}\left(0, g_i^{(j)}\right) \prod_{k\in\partial i\setminus j} \sum_{\sigma_{ki}} m_{k\to\psi_i}\left(0, \sigma_{ki}, g_i^{(j)}\right) + \\
&\quad + m_{i\to\psi_i}\left(t_i^{(j)}, g_i^{(j)}\right) \mathbb{I}\left(\sigma_{ji} = 1, 2\right) \prod_{k\in\partial i\setminus j} \sum_{\sigma_{ki}=1,2} m_{k\to\psi_i}\left(t_i^{(j)}, \sigma_{ki}, g_i^{(j)}\right) \\
&\quad - m_{i\to\psi_i}\left(t_i^{(j)}, g_i^{(j)}\right) \mathbb{I}\left(\sigma_{ji} = 2\right) \prod_{k\in\partial i\setminus j} m_{k\to\psi_i}\left(t_i^{(j)}, 2, g_i^{(j)}\right) \qquad \text{(B.1.3)}
\end{aligned}
$$

Similarly, the outgoing message to the (t_i, g_i) variable node is:

$$p_{\psi_i \to i}(t_i, g_i) \propto \delta(t_i, 0) \prod_{k \in \partial i} \sum_{\sigma_{ki}} m_{k \to \psi_i}(0, \sigma_{ki}, g_i) + \\ + \prod_{k \in \partial i} \sum_{\sigma_{ki}=1,2} m_{k \to \psi_i}(t_i, \sigma_{ki}, g_i) \\ - \prod_{k \in \partial i} m_{k \to \psi_i}(t_i, 2, g_i) \tag{B.1.4}$$

In the simplified (t, σ, g) representation for the messages, the update equation for the ϕ_{ij} nodes reads:

$$p_{\phi_{ij} \to j}(t_j, \sigma_{ij}, g_j) \propto \sum_{t_i, \sigma_{ji}, g_i} \Omega(t_i, t_j, \sigma_{ij}, \sigma_{ji}, g_i, g_j)\, m_{i \to \phi_{ij}}(t_i, \sigma_{ji}, g_i) \tag{B.1.5}$$

where:

$$\Omega(t_i, t_j, \sigma_{ij}, \sigma_{ji}, g_i, g_j) = \begin{cases} \chi(t_i, t_j, \sigma_{ij}, g_i) & : t_i < t_j,\ \sigma_{ji} = 2,\ \sigma_{ij} \neq 2 \\ \chi(t_i, t_j, \sigma_{ij}, g_i) + (1-\lambda)^{g_i+1} & : t_i < t_j,\ \sigma_{ji} = 2,\ \sigma_{ji} = 2 \\ \chi(t_j, t_i, \sigma_{ji}, g_j) & : t_j < t_i,\ \sigma_{ji} = 2,\ \sigma_{ji} \neq 2 \\ \chi(t_j, t_i, \sigma_{ji}, g_j) + (1-\lambda)^{g_j+1} & : t_j < t_i,\ \sigma_{ij} = 2,\ \sigma_{ji} = 2 \\ 1 & : t_i = t_j,\ \sigma_{ji} = \sigma_{ij} = 2 \\ 0 & : otherwise \end{cases} \tag{B.1.6}$$

and

$$\chi(t_1, t_2, \sigma, g) = \sum_{t=t_1}^{t_1+g} \delta(\sigma(t_2, t), \sigma)\, \lambda (1-\lambda)^{t-t_1} \tag{B.1.7}$$

Simple algebra and pre-calculation of terms in (B.1.5)-(B.1.7) brings a significant optimization for updates involving the factor node ϕ_{ij} down to $O(TG^2)$ operations per update. A possible pseudo-code implementation of this optimization is shown below in the Algorithm section.

Algorithm 2 Update of factor ϕ_{ij}. This routine computes the ouput message $p_{\phi_{ij}\to j}(t_j, \sigma_{ij}, g_j)$ from the input message $m_{i\to\phi_{ij}}(t_i, \sigma_{ji}, g_i)$. For simplicity, the value $t = \infty$ is stored in position $T_{inf} = T+1$ on the messages. This update can be used with a transimission probability λ_t than can eventually depend on time.

for $\sigma_i = 0$ **to** 2 **do**
  for $g_i = 0$ **to** G **do**
    for $t_i = 0$ **to** T_{inf} **do**
      $H(t_i, \sigma_i)$ += $m_{i\to\phi_{ij}}(t_i, \sigma_i, g_i)$
    end for
  end for
  $R(T_{inf}, \sigma_i) \leftarrow H(T_{inf}, \sigma_i)$
  for $t = T$ **to** 0 **do**
    $R(t_i, \sigma_i)$ += $H(t_i, \sigma_i)$
  end for
end for
for $g_j = 0$ **to** G **do**
  for $t_j = 0$ **to** T_{inf} **do**
    $t_n \leftarrow \min(t_j + g_j, T)$
    $clear(g0, q0)$
    $p \leftarrow 1$
    for $t = t_j$ **to** t_n **do**
      $q0(t) \leftarrow p$
      $g0(t) \leftarrow g0(t-1) + p \cdot \lambda_t$
      $p \leftarrow p \cdot (1 - \lambda_t)$
    end for
    $q0(t_n + 1) \leftarrow p$
    for $t_i = t_j + 1$ **to** $t_n + 1$ **do**
      $z_0 \leftarrow g0(t_i - 2) - g0(t_j - 1)$
      $z_1 \leftarrow q0(t_i - 1) \cdot \lambda_{t_i - 1}$
      $z_2 \leftarrow q0(t_n + 1) + g0(t_n) - g0(t_i - 1)$
      $p_{\phi_{ij}\to j}(t_j, 2, g_j)$ += $z_0 \cdot H(t_i, 0) + z_1 \cdot H(t_i, 1) + z_2 \cdot H(t_i, 2)$
      $U(t_i, 0)$ += $z_0 \cdot m_{i\to\phi_{ij}}(t_j, 2, g_j)$
      $U(t_i, 1)$ += $z_1 \cdot m_{i\to\phi_{ij}}(t_j, 2, g_j)$
      $U(t_i, 2)$ += $z_2 \cdot m_{i\to\phi_{ij}}(t_j, 2, g_j)$
    end for
    if $t_n + 2 \le T_{inf}$ **then**
      $z_0 \leftarrow g0(t_n)$
      $z_2 \leftarrow q0(t_n + 1)$
      $p_{\phi_{ij}\to j}(t_j, 2, g_j)$ += $z_0 \cdot R(t_n + 2, 0) + z_2 \cdot R(t_n + 2, 2)$
      $S(t_n + 2, 0)$ += $z_0 \cdot m_{i\to\phi_{ij}}(t_j, 2, g_j)$
      $S(t_n + 2, 2)$ += $z_2 \cdot m_{i\to\phi_{ij}}(t_j, 2, g_j)$
    end if
  end for
end for

Algorithm 2 (continued)

```
for g_j = 0 to G do
    for σ_j = 0 to 2 do
        τ ← 0
        for t_j = 0 to T_inf do
            τ += S(t_j, σ_j)
            p_{φ_ij→j}(t_j, σ_j, g_j) += τ + U(t_j, σ_j)
        end for
    end for
end for
for t_j = 0 to T_inf do
    for g_j = 0 to G do
        p_{φ_ij→j}(t_j, 2, g_j) += H(t_j, 2)
    end for
end for
```

B.2. GA method for the inference of the epidemic parameters

The computation of the gradient of the free energy deserves some special attention: being f a function of all the BP messages, one would argue that this messages depend on the model parameters too, at every step in the BP algorithm. Actually, there is no need to consider this implicit (λ, μ) dependence if BP has reached its fixed point, that is when BP equations are satisfied and the messages are nothing else but Lagrange multipliers with respect to the constraint minimization of the Bethe free energy functional [**36**]. In this scheme, the only explicit dependence of free energy on epidemic parameters is in the factor node terms f_a's involving compatibility functions $\phi_{ij} = \omega_{ij}\left(t_{ij} - t_i | g_i\right) \omega_{ji}\left(t_{ji} - t_j | g_j\right)$ and $\mathcal{G}_i\left(g_i\right) = \mu_i \left(1 - \mu_i\right)^{g_i}$, and the gradient can be computed very easily. For the ϕ_{ij} nodes one has:

$$\frac{\partial f_{\phi_{ij}}}{\partial \lambda} = \frac{\sum_{t_i, t_{ji}, g_i, t_j, t_{ij}, g_j} \frac{\partial \phi_{ij}}{\partial \lambda}\left(t_i, t_{ji}, g_i, t_j, t_{ij}, g_j\right) m_{i \to \phi_{ij}}\left(t_i, t_{ji}, g_i\right) m_{j \to \phi_{ij}}\left(t_j, t_{ij}, g_j\right)}{\sum_{t_i, t_{ji}, g_i, t_j, t_{ij}, g_j} \phi_{ij}\left(t_i, t_{ji}, g_i, t_j, t_{ij}, g_j\right) m_{i \to \phi_{ij}}\left(t_i, t_{ji}, g_i\right) m_{j \to \phi_{ij}}\left(t_j, t_{ij}, g_j\right)} \tag{B.2.1}$$

where

$$\frac{\partial \phi_{ij}}{\partial \lambda} = \begin{cases} 1 & t_i < t_j \text{ and } t_i = t_{ij} < t_i + g_i \\ -\left(g_i - t_i\right) \lambda \left(1 - \lambda\right)^{g_i - t_i - 1} & t_i < t_j \text{ and } t_i < t_{ij} = t_i + g_i \\ \left(1 - \lambda\right)^{t_{ij} - t_i} - \left(t_{ij} - t_i\right) \lambda \left(1 - \lambda\right)^{t_{ij} - t_i - 1} & t_i < t_j \text{ and } t_i < t_{ij} < t_i + g_i \\ 1 & t_j < t_i \text{ and } t_j = t_j < t_j + g_j \\ -\left(g_j - t_j\right) \lambda \left(1 - \lambda\right)^{g_j - t_j - 1} & t_j < t_i \text{ and } t_j < t_{ji} = t_j + g_j \\ \left(1 - \lambda\right)^{t_{ji} - t_j} - \left(t_{ji} - t_j\right) \lambda \left(1 - \lambda\right)^{t_{ji} - t_j - 1} & t_j < t_i \text{ and } t_j < t_{ji} < t_j + g_j \\ 0 & \text{else} \end{cases} \tag{B.2.2}$$

In the simplified (t, σ, g) representation for the messages, equation (B.2.2) takes the form:

$$\frac{\partial \phi_{ij}}{\partial \lambda} = \begin{cases} \chi\left(t_i, t_j, \sigma_{ij}, g_i\right) & t_i < t_j, \sigma_{ji} = 2, \sigma_{ij} \neq 2 \\ \chi\left(t_i, t_j, \sigma_{ij}, g_i\right) - \left(g_i + 1\right)\left(1 - \lambda\right)^{g_i} & t_i < t_j, \sigma_{ji} = 2, \sigma_{ij} = 2 \\ \chi\left(t_j, t_i, \sigma_{ji}, g_j\right) & t_j < t_i, \sigma_{ji} = 2, \sigma_{ij} \neq 2 \\ \chi\left(t_j, t_i, \sigma_{ji}, g_j\right) - \left(g_j + 1\right)\left(1 - \lambda\right)^{g_j} & t_j < t_i, \sigma_{ji} = 2, \sigma_{ij} = 2 \\ 0 & otherwise \end{cases} \tag{B.2.3}$$

where:

$$\chi\left(t_1, t_2, \sigma, g\right) = \sum_{t=t_1}^{t_1+g} \delta\left(\sigma\left(t_2, t\right), \sigma\right)\left(1 - \lambda\right)^{t-t_1} - \left(t - t_1\right)\lambda\left(1 - \lambda\right)^{t-t_1-1} \tag{B.2.4}$$

For the $\mathcal{G}_i$ nodes the gradient reads:

$$\frac{\partial f_{\mathcal{G}_i}}{\partial \mu} = \frac{\sum_{g_i} \tilde{\mathcal{G}}_i(g_i) m_{i \to \mathcal{G}_i}(g_i)}{\sum_{g_i} \mathcal{G}_i(g_i) m_{i \to \mathcal{G}_i}(g_i)} \tag{B.2.5}$$

where

$$\tilde{\mathcal{G}}_i(g_i) = \begin{cases} (1-\mu)^{g_i} - g_i \mu (1-\mu)^{g_i - 1} & : g_i < G \\ G - G(1-\mu)^{G-1} & : g_i = G. \end{cases} \tag{B.2.6}$$

APPENDIX C

Inverse Dynamics in continuous time: efficient BP updates

C.1. Efficient BP updates for inference on SIR in continuous time contact networks

The BP equations for the factor node ψ_i are:

$$\begin{aligned}
P_{ij}\left(s_{ij},s_{ji}\right) \quad \propto \quad & \sum_{t_i} P_{ii}\left(t_i\right) \int_0^\infty dg_i G\left(g_i\right) \sum_{r_{ij}} R\left(r_{ij}\right) \delta\left(s_{ij}, S_{ij}\left(t_i, r_{ij}, g_i\right)\right) \times \\
& \times \sum_{\{s_{ki}\}} \delta\left(t_i, \min_{k\in\partial i} T_{ki}\left(s_{ki}\right)\right) \prod_{k\in\partial i\backslash j} \left\{ \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right) \right\} \\
= \quad & \sum_{t_i<T_{ji}(s_{ji})} P_{ii}\left(t_i\right) \int_0^\infty dg_i G\left(g_i\right) \sum_{r_{ij}} R\left(r_{ij}\right) \delta\left(s_{ij}, S_{ij}\left(t_i, r_{ij}, g_i\right)\right) \times \\
& \times \Bigg\{ \prod_{k\in\partial i\backslash j} \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})\geq t_i} P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right) - \\
& - \prod_{k\in\partial i\backslash j} \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})>t_i} P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right) \Bigg\} \\
& + P_{ii}\left(T_{ji}\left(s_{ji}\right)\right) \int_0^\infty dg_i G\left(g_i\right) \sum_{r_{ij}} R\left(r_{ij}\right) \delta\left(s_{ij}, S_{ij}\left(T_{ji}\left(s_{ji}\right), r_{ij}, g_i\right)\right) \times \\
& \times \prod_{k\in\partial i\backslash j} \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})\geq t_i} P_{ki}\left(s_{ki}, S_{ik}\left(T_{ji}\left(s_{ji}\right), r_{ik}, g_i\right)\right)
\end{aligned} \tag{C.1.1}$$

and for

$$\begin{aligned}
P_i\left(t_i\right) \quad \propto \quad & \int dg_i G\left(g_i\right) \Bigg\{ \prod_{k\in\partial i} \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})\geq t_i} P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right) - \\
& - \prod_{k\in\partial i} \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})>t_i} P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right) \Bigg\}
\end{aligned} \tag{C.1.2}$$

The definite integrals $\int dg_i G\left(g_i\right)$ over the probability distribution of the recovery times can be pre-computed via the quantities $K_r = \int_{h_r}^{h_{r+1}} G\left(g_i\right) dg_i$, and essentially proceed as if g_i took only discrete values $\frac{1}{2}\left(h_r + h_{r+1}\right)$ with probabilities given by K_r. If node i has a total of n contacts, a naive implementation of the resulting update has complexity $O\left(n^2 d\right)$.

All terms $Q_{ki}^{\geq}\left(t_i, g_i\right) = \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})\geq t_i} P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right)$ and $Q_{ki}^{>}\left(t_i, g_i\right) = \sum_{r_{ik}} R_{ik}\left(r_{ik}\right) \sum_{s_{ki}:T_{ki}(s_{ki})\geq t_i} P_{ki}\left(s_{ki}, S_{ik}\left(t_i, r_{ik}, g_i\right)\right)$ can be simultaneously computed in time $O\left(n^2\right)$.

Then

$$P_i(t_i) \propto \sum_{t_i<T_{ji}(s_{ji})} P_{ii}(t_i) \int_0^\infty dg_i G(g_i) \sum_{r_{ij}} R(r_{ij}) \times \\ \times \delta(s_{ij}, S_{ij}(t_i, r_{ij}, g_i)) \left(\prod_{k\in\partial i\setminus j} Q^{\geq}_{ki}(t_i, g_i) - \prod_{k\in\partial i\setminus j} Q^{>}_{ki}(t_i, g_i) \right)$$

can be computed on a single loop over the possible values of t_i, g_i.

Bibliography

[1] Marc Mézard, Giorgio Parisi, and Miguel Virasoro. *Spin Glass Theory and Beyond*. World Scientific Lecture Notes in Physics, 1987.

[2] Marc Mézard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, January 2009.

[3] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[5] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[6] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *ArXiv e-prints*, December 2013.

[7] Elizabeth Gardner and Bernard Derrida. Three unfinished works on the optimal storage capacity of networks. *J. Phys. A: Math. Gen.*, 22:1983–1996, 1989.

[8] Hanoch Gutfreund and Yaakov Stein. Capacity of neural networks with discrete synaptic couplings. *Journal of Physics A: Mathematical and General*, 23(12):2613, 1990.

[9] Holm Schwarze and John Hertz. Discontinuous generalization in large committee machines. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 399–406. Morgan-Kaufmann, 1994.

[10] David Saad (editor). *On-Line Learning in Neural Networks*. Publications of the Newton Institute, 1998.

[11] David Saad and Sara A. Solla. Exact solution for on-line learning in multilayer neural networks. *Phys. Rev. Lett.*, 74:4337–4340, May 1995.

[12] Daniel J Amit. *Modeling Brain Functions: The world of attractor neural networks*. Cambridge University Press, 1992.

[13] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[14] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[15] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, Alejandro Lage-Castellanos, and Riccardo Zecchina. Bayesian inference of epidemics on networks via belief propagation. *Physical Review Letters*, 112(11):118701, March 2014.

[16] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35:1792–1796, Dec 1975.

[17] Elizabeth Gardner. The space of interactions in neural network models. *Phys. Rev. A*, (42), 1988.

[18] S F Edwards and P W Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5(5):965, 1975.

[19] G. Parisi. Infinite number of order parameters for spin-glasses. *Phys. Rev. Lett.*, 43:1754–1756, Dec 1979.

[20] Daniel J. Amit, Hanoch Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.*, 55:1530–1533, Sep 1985.

[21] Daniel J. Amit, Hanoch Gutfreund, and H. Sompolinsky. Spin-glass models of neural networks. *Phys. Rev. A*, 32:1007–1018, Aug 1985.

[22] T.M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *Electronic Computers, IEEE Transactions on*, EC-14(3):326–334, June 1965.

[23] Elizabeth Gardner and Bernard Derrida. Optimal storage properties of neural network models. (22), 1989.

[24] Werner Krauth and Marc Mézard. Storage capacity of memory networks with binary couplings. *J. Phys. France*, 50:3057–3066, 1989.

[25] W Krauth and M Opper. Critical storage capacity of the j = +1,-1 neural network. *Journal of Physics A: Mathematical and General*, 22(11):L519, 1989.

[26] B Derrida, R B Griffiths, and A Prugel-Bennett. Finite-size effects and bounds for perceptron models. *Journal of Physics A: Mathematical and General*, 24(20):4907, 1991.
[27] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
[28] J.S. Denker, D. Schwarz, B. Wittner, S. Solla, R. Howard, and L. Jackel. Large automatic learning, rule extraction, and generalization. (1), 1987.
[29] A. Priel, M. Blatt, T. Grossmann, E. Domany, and I. Kanter. Computational capabilities of restricted two-layered perceptrons. (50), 1994.
[30] G.R. Shorack and J.A. Wellner. *Empirical processes with applications to Statistics*. Classics in Applied Mathematics, 1986.
[31] G.J. Mitchison and R.M. Durbin. Bounds on the learning capacity of some multi-layer networks. *Biological Cybernetics*, 60(5):345–365, 1989.
[32] Manfred Opper. Statistical physics estimates for the complexity of feedforward neural networks. *Phys. Rev. E*, 51:3613–3618, Apr 1995.
[33] R. Monasson and D. O'Kane. Domains of solutions and replica symmetry breaking in multilayer neural networks. *EPL (Europhysics Letters)*, 27(2):85, 1994.
[34] Rémi Monasson and Riccardo Zecchina. Weight space structure and internal representations: A direct approach to learning and generalization in multilayer neural networks. *Phys. Rev. Lett.*, 76:2205–2205, Mar 1996.
[35] Rémi Monasson and Riccardo Zecchina. Learning and generalization theories of large committee-machines. *Modern Physics Letters B*, 09(30):1887–1897, 1995.
[36] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Bethe free energy, kikuchi approximations, and belief propagation algorithms. *Advances in neural information processing systems*, 13, 2001.
[37] Mohsen Bayati, Devavrat Shah, and Mayank Sharma. Max-product for maximum weight matching: Convergence, correctness, and LP duality. *IEEE Transactions on Information Theory*, 54(3):1241–1251, March 2008.
[38] Mohsen Bayati, A. Braunstein, and Riccardo Zecchina. A rigorous analysis of the cavity equations for the minimum spanning tree. *Journal of Mathematical Physics*, 49(12):125206, 2008. Cited by 0012.
[39] David Gamarnik, Devavrat Shah, and Yehua Wei. *Belief Propagation for Min-Cost Network Flow: Convergence & Correctness*. Society for Industrial and Applied Mathematics, January 2010. National Science Foundation (U.S.) (Project CMMI-0726733).
[40] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, October 2001.
[41] Mohsen Bayati and Chandra Nair. A rigorous proof of the cavity method for counting matchings. *arXiv preprint cond-mat/0607290*, 2006.
[42] Daniel H. O'Connor, Gayle M. Wittenberg, and Samuel S.-H. Wang. Graded bidirectional synaptic plasticity is composed of switch-like unitary events. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9679–9684, 2005.
[43] Thomas M Bartol, Cailey Bromer, Justin P Kinney, Michael A Chirillo, Jennifer N Bourne, Kristen M Harris, and Terrence J Sejnowski. Hippocampal spine head sizes are highly precise. *bioRxiv*, 2015.
[44] Edoardo Amaldi. On the complexity of training perceptrons. *Kohonen et al*, pages 55–60, 1991.
[45] Haim Sompolinsky, Naftali Tishby, and H. Sebastian Seung. Learning from examples in large neural networks. *Physical Review Letters*, 65(13):1683, 1990.
[46] Haiping Huang, K. Y. Michael Wong, and Yoshiyuki Kabashima. Entropy landscape of solutions in the binary perceptron problem. *Journal of Physics A: Mathematical and Theoretical*, 46(37):375002, 2013.
[47] Tomoyuki Obuchi and Yoshiyuki Kabashima. Weight space structure and analysis using a finite replica number in the ising perceptron. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(12):P12014, 2009.
[48] Heinz Horner. Dynamics of learning for the binary perceptron problem. *Zeitschrift für Physik B Condensed Matter*, 86(2):291–308, 1992.
[49] Haiping Huang and Yoshiyuki Kabashima. Origin of the computational hardness for learning with binary synapses. *Physical Review E*, 90(5):052813, 2014.
[50] Olivier C Martin, Rémi Monasson, and Riccardo Zecchina. Statistical mechanics methods and phase transitions in optimization problems. *Theoretical computer science*, 265(1):3–67, 2001.
[51] Cristopher Moore and Stephan Mertens. *The nature of computation*. Oxford University Press, 2011.
[52] Alfredo Braunstein and Riccardo Zecchina. Learning by message-passing in neural networks with material synapses. *Phys. Rev. Lett.*, 96:030201, 2006.

[53] Carlo Baldassi, Alfredo Braunstein, Nicolas Brunel, and Riccardo Zecchina. Efficient supervised learning in networks with binary synapses. *Proceedings of the National Academy of Sciences*, 104:11079–11084, 2007.

[54] Carlo Baldassi. Generalization learning in a perceptron with binary synapses. *J. Stat. Phys.*, 136:1572, 2009.

[55] Carlo Baldassi and Alfredo Braunstein. A max-sum algorithm for training discrete neural networks (submitted for publication). 2015.

[56] Silvio Franz and Giorgio Parisi. Recipes for metastable states in spin glasses. *Journal de Physique I*, 5(11):1401–1415, 1995.

[57] Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Subdominant dense clusters allow for simple learning and high computational performance in neural networks with discrete synapses. *Phys. Rev. Lett.*, 115:128101, Sep 2015.

[58] C. Baldassi, F. Gerace, C. Lucibello, L. Saglietti, and R. Zecchina. Learning may need only a few bits of synaptic precision. *arXiv preprint arXiv:1602.04129*, February 2016.

[59] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400(6740):133–137, 1999.

[60] Marc Mézard and Riccardo Zecchina. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Physical Review E*, 66(5):056126, 2002.

[61] Marc Mézard, Giorgio Parisi, and Riccardo Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[62] Andrea Montanari, Federico Ricci-Tersenghi, and Guilhem Semerjian. Clusters of solutions and replica symmetry breaking in random k-satisfiability. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(04):P04004, 2008.

[63] Florent Krzakala, Andrea Montanari, Federico Ricci-Tersenghi, Guilhem Semerjian, and Lenka Zdeborova. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

[64] Carlo Baldassi, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Local entropy as a measure for sampling solutions in constraint satisfaction problems (accepted for publication). *Journal of Statistical Mechanics: Theory and Experiment*, 2016.

[65] Devavrat Shah and Tauhid Zaman. Detecting sources of computer viruses in networks: theory and experiment. *ACM SIGMETRICS Performance Evaluation Review*, 38(1):203–214, 2010.

[66] Cesar Henrique Comin and Luciano da Fontoura Costa. Identifying the starting point of a spreading process in complex networks. *Physical Review E*, 84(5):056105, November 2011.

[67] Devavrat Shah and Tauhid Zaman. Rumors in a network: Who's the culprit? *Information Theory, IEEE Transactions on*, 57(8):5163–5181, 2011.

[68] Vincenzo Fioriti and Marta Chinnici. Predicting the sources of an outbreak with a spectral technique. *arXiv preprint arXiv:1211.2333*, 2012.

[69] Pedro C. Pinto, Patrick Thiran, and Martin Vetterli. Locating the source of diffusion in large-scale networks. *Physical Review Letters*, 109(6):068702, August 2012.

[70] Nino Antulov-Fantulin, Alen Lancic, Hrvoje Stefancic, Mile Sikic, and Tomislav Smuc. Statistical inference framework for source detection of contagion processes on arbitrary network structures. March 2013.

[71] WenXiang Dong, Wenyi Zhang, and Chee Wei Tan. Rooting out the rumor culprit from suspects. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2671–2675, July 2013.

[72] Wuqiong Luo, Wee Peng Tay, and Mei Leng. Identifying infection sources and regions in large networks. *IEEE Transactions on Signal Processing*, 61(11):2850–2865, 2013.

[73] Kai Zhu and Lei Ying. Information source detection in the SIR model: A sample path based approach. In *Information Theory and Applications Workshop (ITA), 2013*, pages 1–9, 2013.

[74] N. Karamchandani and M. Franceschetti. Rumor source detection under probabilistic sampling. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2184–2188, July 2013.

[75] Andrey Y. Lokhov, Marc Mézard, Hiroki Ohta, and Lenka Zdeborová. Inferring the origin of an epidemic with a dynamic message-passing algorithm. *Phys. Rev. E*, 90:012801, Jul 2014.

[76] Norman T. J Bailey. *The mathematical theory of infectious diseases and its applications*. Griffin, London, 1975.

[77] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A*, 115(772):700–721, August 1927.

[78] Brian Karrer and M. E. J. Newman. Message passing approach for general epidemic models. *Physical Review E*, 82(1):016101, July 2010.

[79] Chris Milling, Constantine Caramanis, Shie Mannor, and Sanjay Shakkottai. Detecting epidemics using highly noisy data. In *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, page 177–186. ACM, 2013.

[80] Eli A Meirom, Chris Milling, Constantine Caramanis, Shie Mannor, Ariel Orda, and Sanjay Shakkottai. Localized epidemic detection in networks with overwhelming noise. *arXiv preprint arXiv:1402.1263*, 2014.

[81] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, page 137–146, New York, NY, USA, 2003. ACM.

[82] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, and Riccardo Zecchina. Large deviations of cascade processes on graphs. *Physical Review E*, 87(6):062115, June 2013.

[83] F. Altarelli, A. Braunstein, L. Dall'Asta, and R. Zecchina. Optimizing spread dynamics on graphs by message passing. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(09):P09011, September 2013.

[84] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[85] Fabrizio Altarelli, Alfredo Braunstein, Luca Dall'Asta, Alessandro Ingrosso, and Riccardo Zecchina. The patient-zero problem with noisy observations. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(10):P10016, 2014.

[86] Luis E. C. Rocha, Fredrik Liljeros, and Petter Holme. Information dynamics shape the sexual networks of internet-mediated prostitution. *Proceedings of the National Academy of Sciences*, 107(13):5706–5711, March 2010.

[87] Luis E. C. Rocha, Fredrik Liljeros, and Petter Holme. Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. *PLoS Comput Biol*, 7(3):e1001109, March 2011.

[88] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166–180, February 2011.

[89] Leon Danon, Ashley P Ford, Thomas House, Chris P Jewell, Matt J Keeling, Gareth O Roberts, Joshua V Ross, and Matthew C Vernon. Networks and the epidemiology of infectious disease. *Interdisciplinary perspectives on infectious diseases*, 2011, 2011.

[90] Madhav Marathe and Anil Kumar S. Vullikanti. Computational epidemiology. *Commun. ACM*, 56(7):88–96, July 2013.

[91] Alfredo Braunstein and Alessandro Ingrosso. Inference of causality in epidemics on temporal contact networks (in preparation). 2016.

[92] Marcel Salathé, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W. Feldman, and James H. Jones. A high-resolution human contact network for infectious disease transmission. *Proceedings of the National Academy of Sciences*, 107(51):22020–22025, 2010.

[93] Alfredo Braunstein and Alessandro Ingrosso. (in preparation). 2016.

[94] Wayne W. Zachary. An information flow model for conflict and fission in small groups. 33, 1977.

[95] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures & Algorithms*, 27(2):201–226, 2005.